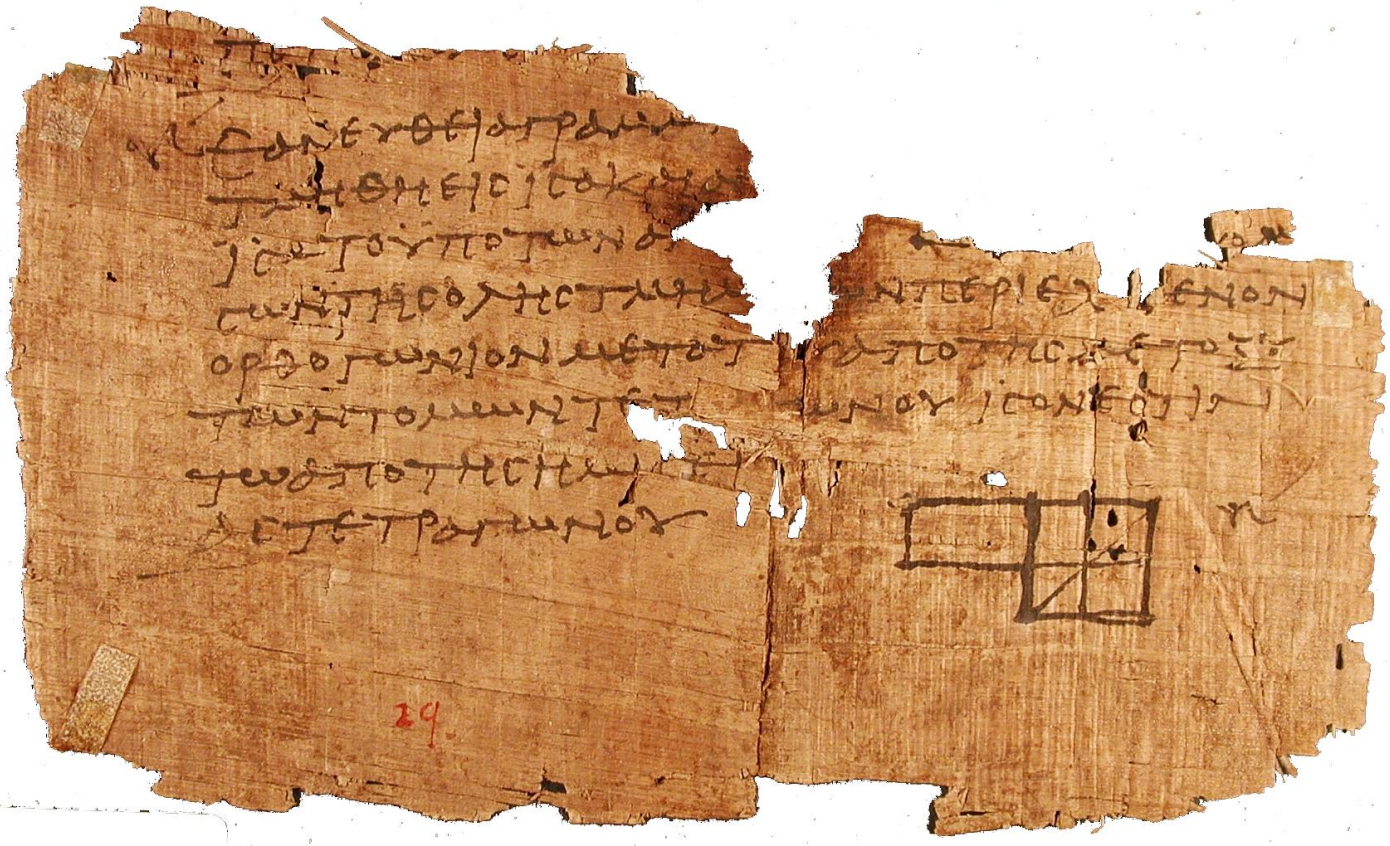
How to prove with zero knowledge

Zero-knowledge proofs primer

Michał Zając
Clearmαtics Ltd / University of Tartu

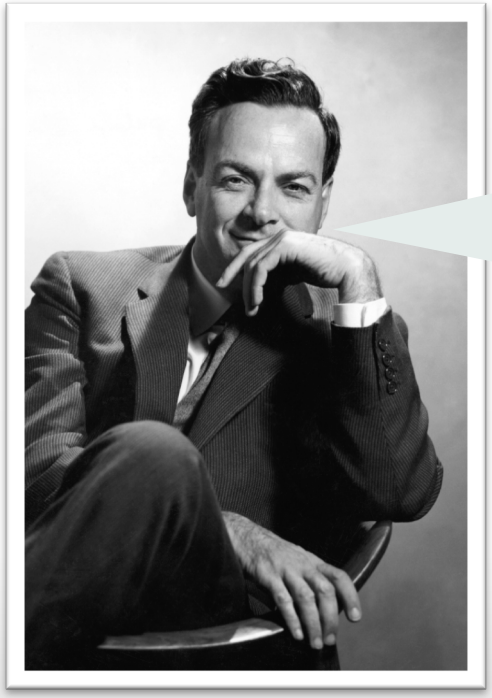
clearmαtics

UNIVERSITY OF TARTU



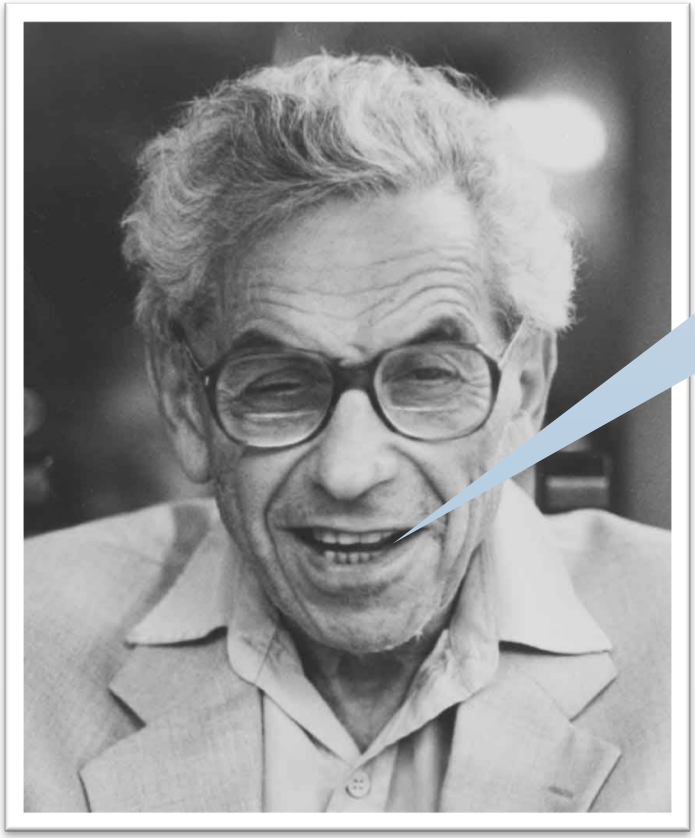
What is proof?

A rigorous mathematical argument which unequivocally demonstrates the truth of a given proposition. A mathematical statement that has been proven is called a **theorem**.



Any theorem, no matter how difficult to prove in the first place, is viewed as trivial by mathematicians once it has been proven. Therefore, there are exactly two types of mathematical objects: trivial ones, and those which have not yet been proven.

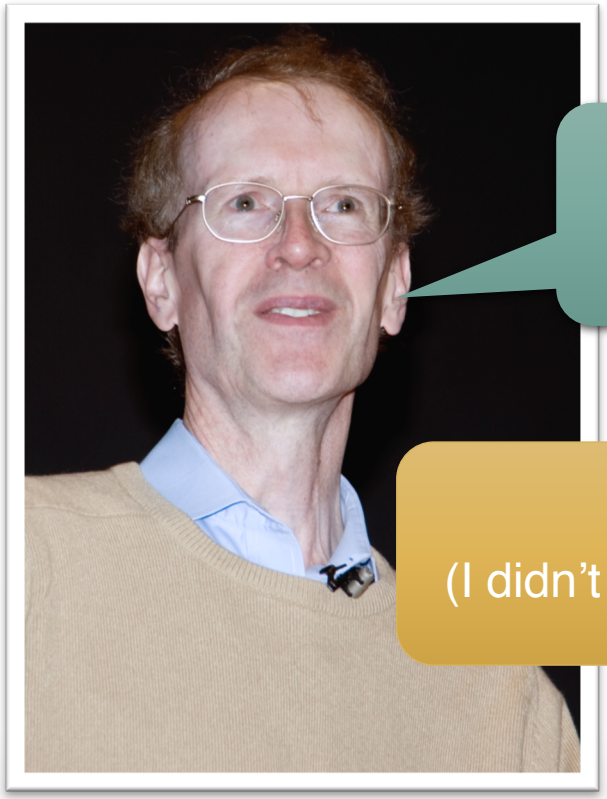
How do we prove?



A mathematician is a machine for converting coffee* into theorems

* - weak coffee is suitable only for lemmas

How we prove?



I know how to prove
Fermat's last theorem

Show me!
(I didn't know it was my last ☹)

How we prove?

Come to my lecture

Oxford Mathematics
London Public Lecture

28.11.17

UNIVERSITY OF OXFORD
Mathematical Institute

ew

Wiles

Oxford Mathematics in partnership with the Science Museum is delighted to announce its first Public Lecture in London. World-renowned mathematician Andrew Wiles will be our speaker. Andrew will be talking about his current work and will also be in conversation with mathematician and broadcaster Hannah Fry after the lecture.

Please email external-relations@maths.ox.ac.uk to attend.

6.30pm Tuesday 28 November 2017
Science Museum, London SW7 2DD

Oxford
Mathematics

What Fermat learns?

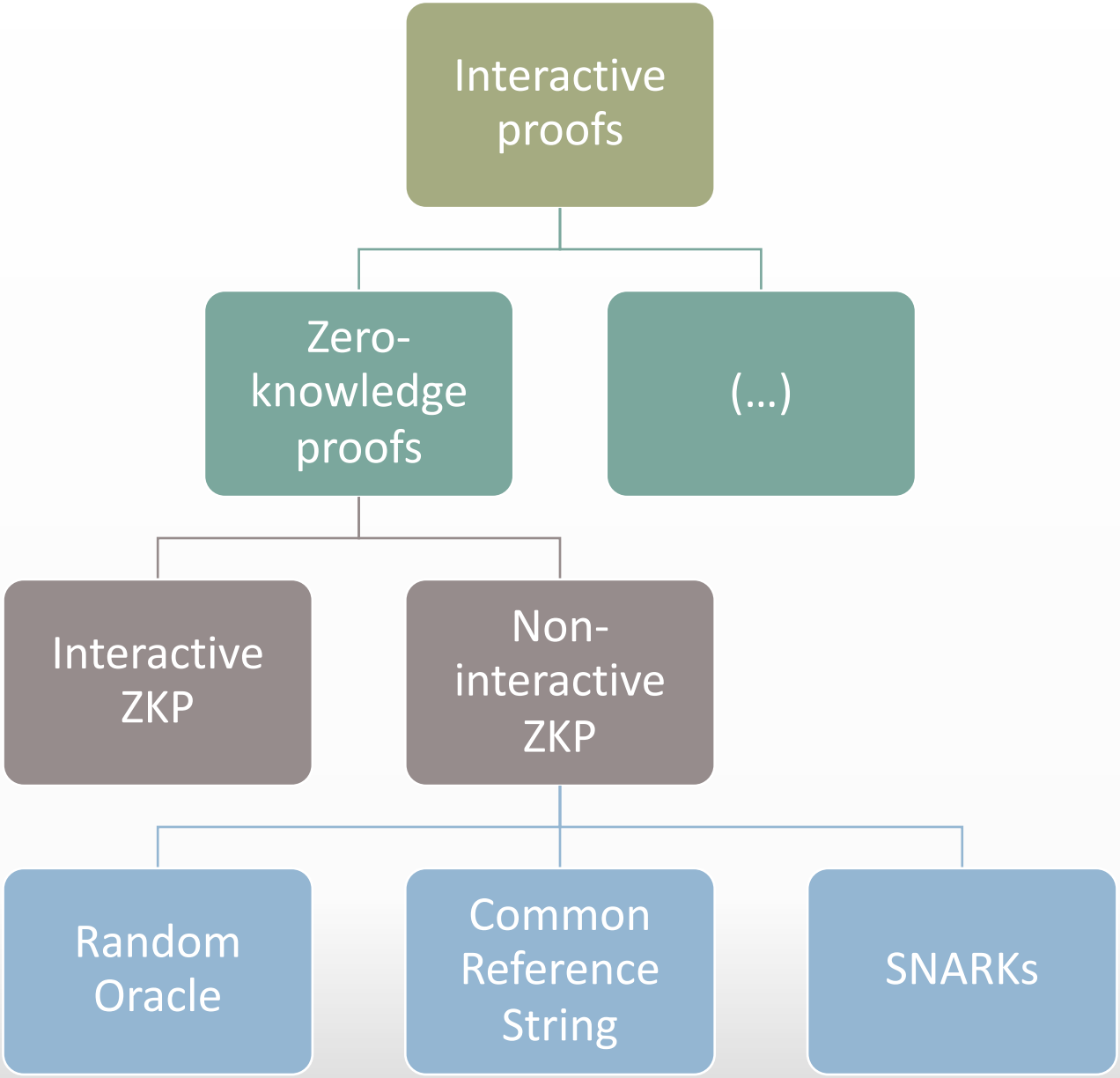
- His theorem was true
- Proof of the theorem
- **He could later convince other folks that he invented the proof!**

Could Andrew convince Pierre that he knows the proof without revealing it?

Could Andrew convince Pierre that he knows the proof without revealing it?

He can

Proof that reveals nothing but the fact the statement holds is called a **zero-knowledge proof**


Interactive proofs
Zero-knowledge proofs
(…)
Interactive ZKP
Non-interactive ZKP
Random Oracle
Common Reference String
SNARKs

A few definitions from complexity theory

Language $\mathcal{L}$ is a set of **words** x that has some property, e.g. composite numbers, words with 'a', graph isomorphic to a given graph G

We say that a language $\mathcal{L}$ is recognisable in time T if there is an algorithm $\mathcal{M}$ that for all $x \in \mathcal{L}$ returns **ACCEPT** in time T

We say $\mathcal{M}$ works in **polynomial time** if for every input x it stops in time bounded by some polynomial $poly(|x|)$, where $|x|$ is the length of x

In cryptography all times are given regarding the **security parameter** n, usually $n \approx 128$

Languages and statements

$$QR_N = \{x \mid x \text{ is quadratic residure mod } N\}$$

There exists machine M such that for all $x \in QR_n$
M says **ACCEPT**

$$GI_G = \{H \mid H \text{ is isomorphic to } G\}$$

There exists machine M such that for all $x \in GI_G$
M says **ACCEPT**

$$\overline{GI}_G = \{H \mid H \text{ is } \boldsymbol{not} \text{ isomorphic to } G\}$$

There exists machine M such that for all $x \in \overline{GI}_G$
M says **ACCEPT**

$$MAT = \{x \mid x \text{ is a valid mathematical theorem}\}$$

$\mathbb{P}$ and $\mathbb{NP}$

Language $\mathcal{L}$ belongs to class $\mathbb{P}$ if there exists machine $\mathcal{M}$ that recognises $\mathcal{L}$ in polynomial time

$$\mathcal{L} = \{x \mid \mathcal{M}(x) = ACCEPT\}$$

Language $\mathcal{L}$ belongs to class $\mathbb{NP}$ if there exists machine $\mathcal{M}$ that for all $x \in \mathcal{L}$ there exists w (of length $poly(|x|)$) and $\mathcal{M}(x, w) = 1$ in polynomial time

Language $\mathcal{L}$ belongs to class $\mathbb{BPP}$ if there exists probabilistic machine $\mathcal{M}$ such that

- $\forall x \in \mathcal{L}, \Pr_R[\mathcal{M}(x) = 1] \geq \frac{2}{3}$
- $\forall x \notin \mathcal{L}, \Pr_R[\mathcal{M}(x) = 1] \leq \frac{1}{3}$

$\mathbb{NP}$-problems

Exemplary $\mathbb{NP}$ problems

Problem is in $\mathbb{NP}$ if we can efficiently verify its solutions

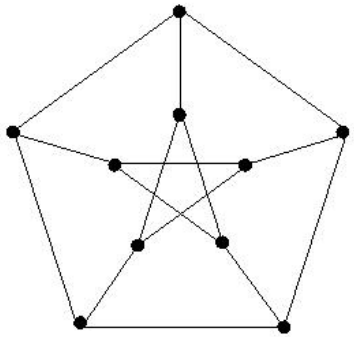
Factorization of integers:

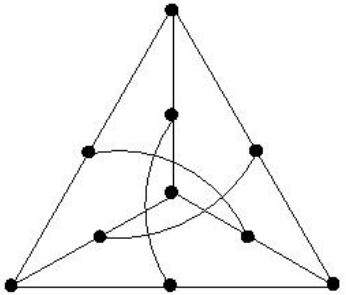
- No general factorization algorithm
- Given factorization we can check

Factorize: $71\,850\,192\,453$ (hard)
Check that:
71850192453 = 47*39*23*141*237*3*17 (easy)

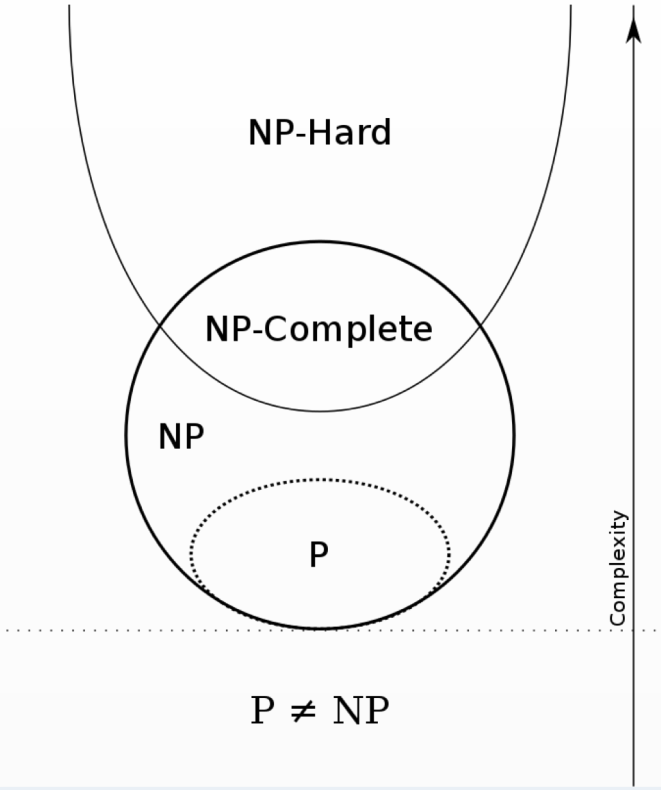
Graph Isomorphism

- Hard to tell whether two graphs are isomorphic
- Given the isomorphism itself we can check





$\mathbb{NP}$-class



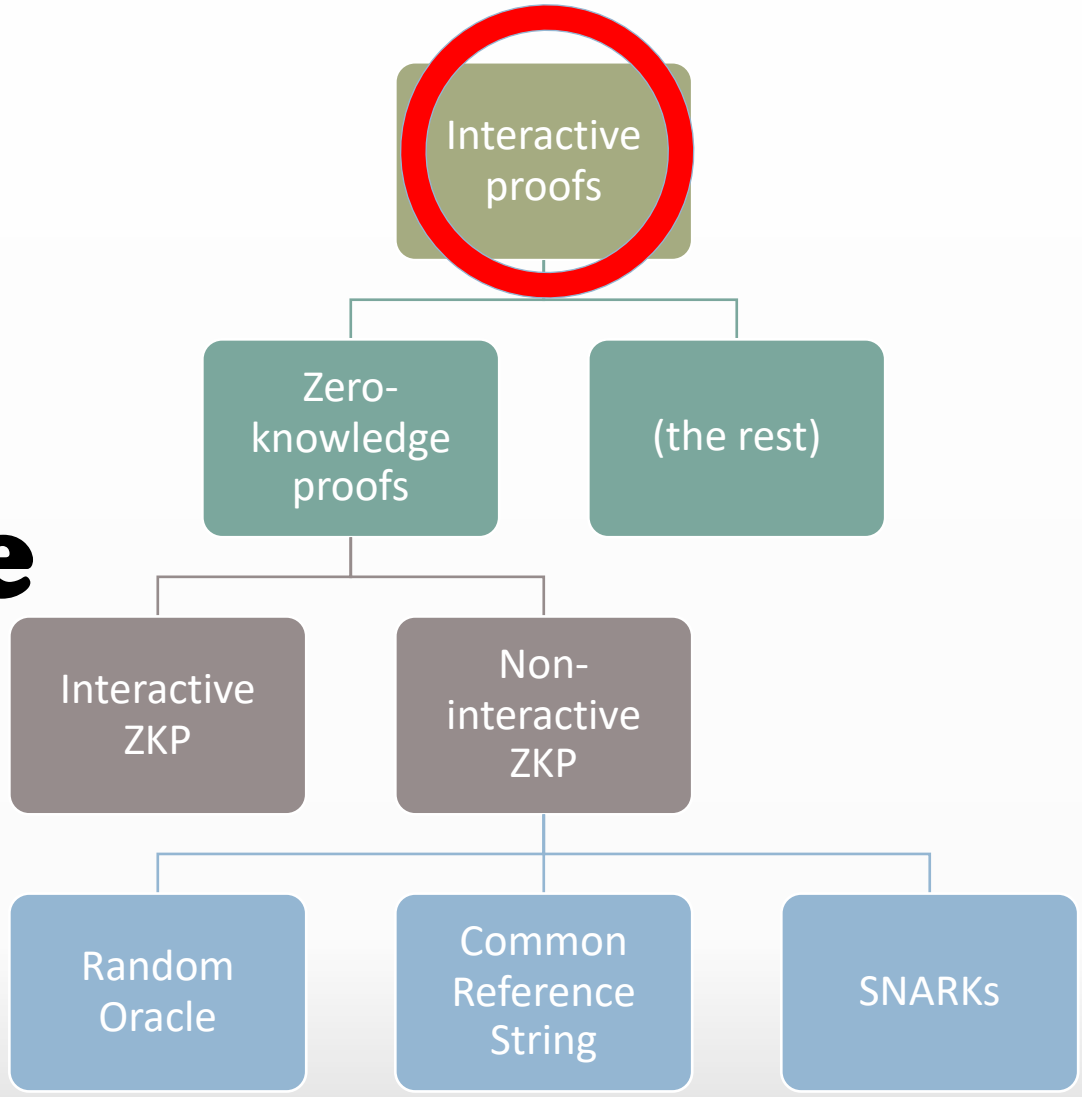
All problems in $\mathbb{P}$ are in $\mathbb{NP}$

$\mathbb{NP}$-hard problems:
Problems at least as hard as any problems in class $\mathbb{NP}$

$\mathbb{NP}$-complete problems:
If we can solve *one* of them, we can solve all problems in $\mathbb{NP}$
(all $\mathbb{NP}$ problems can be **reduced** to an $\mathbb{NP}$-complete problem)

If we could proof that an $\mathbb{NP}$-complete problem can be proven in zero knowledge, then all $\mathbb{NP}$ problems can be solved in zero knowledge!

Interactive proofs



Interactive proof

Prover $\mathcal{P}$
x, w

Prover interacts with **Verifier** convincing him that the proposition is true

Verifier $\mathcal{V}$
x

α_1

β_1

...

α_n

ACCEPT $x \in \mathcal{L}$

REJECT $x \notin \mathcal{L}$

Interactive proof

Prover $\mathcal{P}$
x, w

I know that $x \in \mathcal{L}$ so I want Pierre to accept my proof

Is it possible that I accept a proof but $x \notin \mathcal{L}$?

Verifier $\mathcal{V}$
x

Interactive proof

Definition: An **interactive proof system** for membership in $\mathcal{L}$ is a pair of algorithms (P, V) such that $\forall x$:

COMPLETENESS:

If $x \in \mathcal{L}$, then $\Pr_{R}[(P(w), V)(x) = \boldsymbol{ACCEPT}] \geq \frac{2}{3}$

SOUNDNESS:

If $x \notin \mathcal{L}$, then $\Pr_{R}[(P^*, V)(x) = \boldsymbol{ACCEPT}] \leq \frac{1}{3}$

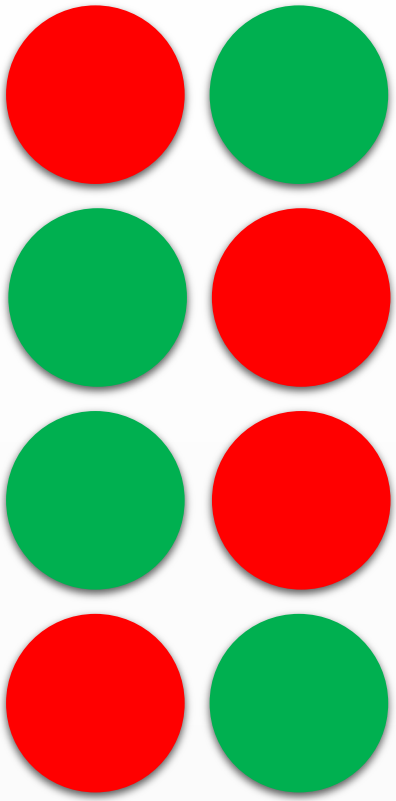
Probabilistic nature of the proof

- It need to be repeated n times to get soundness around $\left(\frac{1}{2}\right)^n$
  - If P^* is **very** lucky he can convince V to accept $x \notin \mathcal{L}$

Color blindness

P

You did!

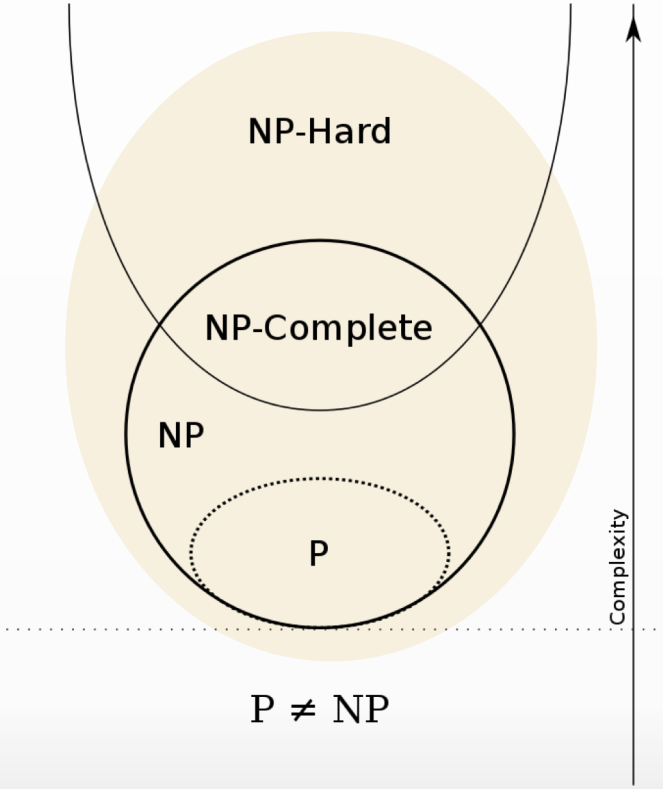


V

Did I swap?

The procedure is randomized by **V**'s randomness
In fact, **V** doesn't need **P** to play!

What can we prove interactively?



Class of problems provable interactively: $\mathbb{IP}$

$\mathbb{P}$-space: problems that can be solved in polynomial **memory** (don't care about the time!)

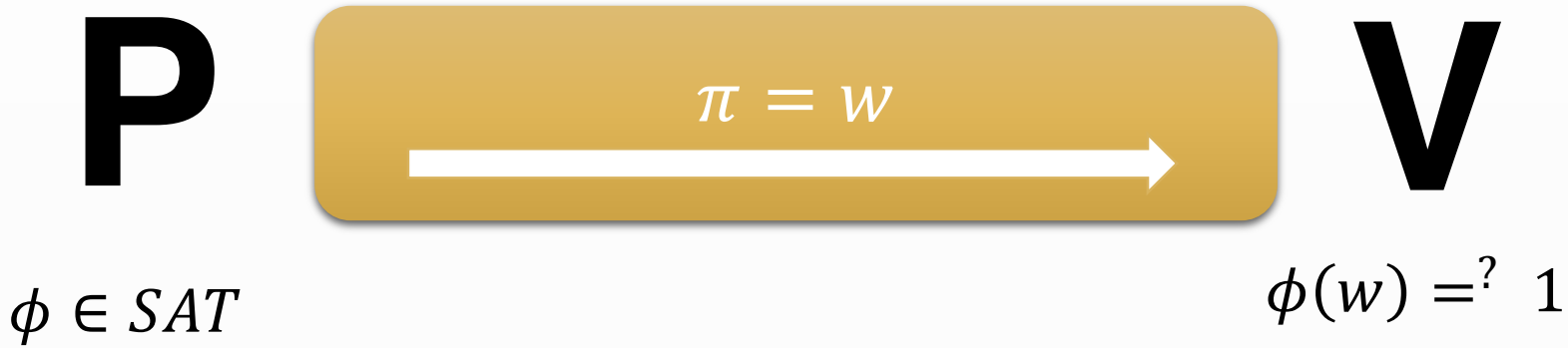
$\mathbb{NP} \subset \mathbb{P}$-space

$\mathbb{IP} \subset \mathbb{P}$-space

Boolean satisfiability

$$SAT = \{\phi \mid \phi \; is \; satisfiable \; boolean \; formula\}$$

$$SAT = \{\phi(w_1, \dots, w_n) \mid \exists \, w \in \{0,1\}^n \, , \phi(w) = 1\}$$



COMPLETE? – **YES** since **V** accepts if $\phi \in SAT$

SOUND? – **YES** since **V** rejects if $\phi \notin SAT$

Quadratic residuosity

$$QR_N = \{x \mid x \text{ is quadratic residure mod } N\}$$

P → $\pi = w$ → **V**

$x \in QR$

$w: w^2 = x$

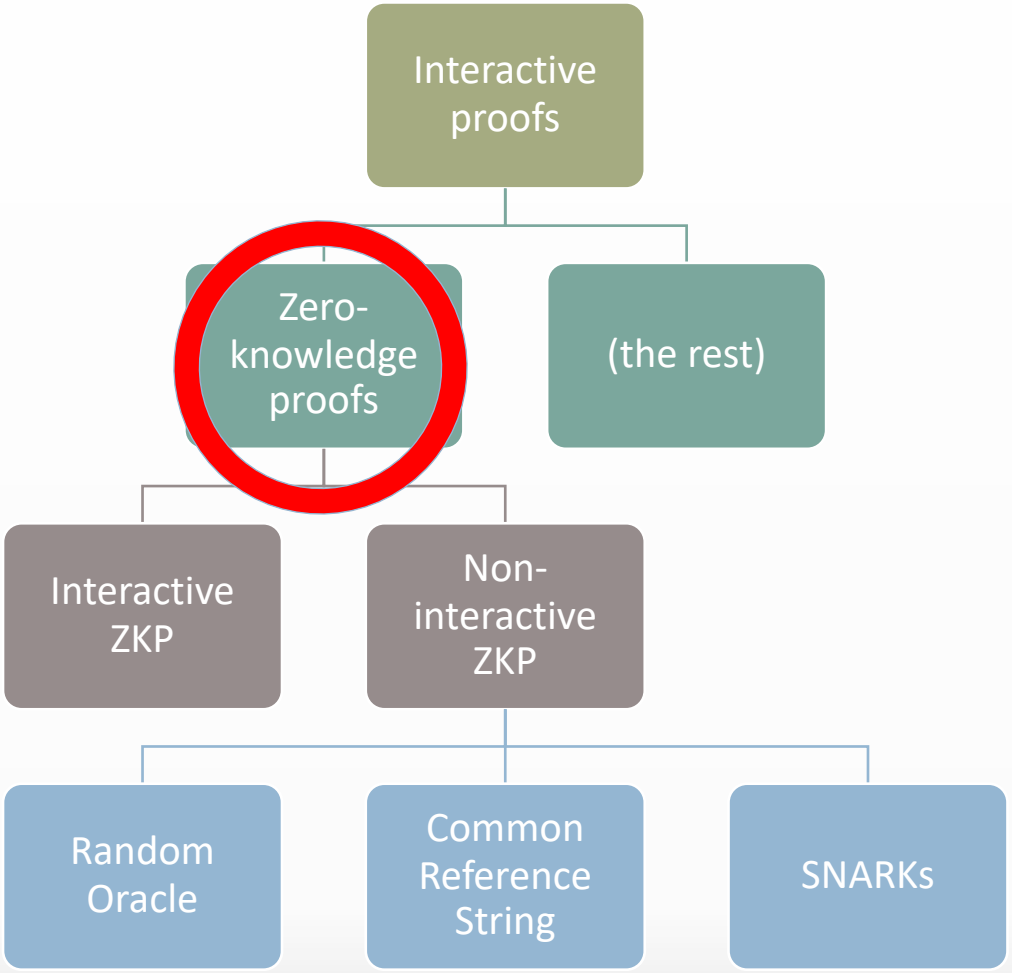
$w^2 =^? x \bmod N?$

V got something for free from seeing π

V might not be able to find w on his own!

Zero knowledge

How to make **V** learn nothing



Zero-knowledge proof

Prover $\mathcal{P}$
x, w

I want Pierre to believe me without showing w

Verifier $\mathcal{V}$
x

How to define zero knowledge?

- **V** didn't learn w
- **V** didn't learn any symbol of w
- **V** didn't learn any information about w
- **V** didn't learn **anything** except $x \in \mathcal{L}$

When **V** **did** learn something?

If **V** can compute something he couldn't computed before

Zero knowledge: whatever is computed following interaction could been computed without it

Zero knowledge

V's view = **V**'s random coins and messages it receives

Zero knowledge:
if for all $x \in \mathcal{L}$ **V**'s view can be efficiently **simulated**

What does it mean?

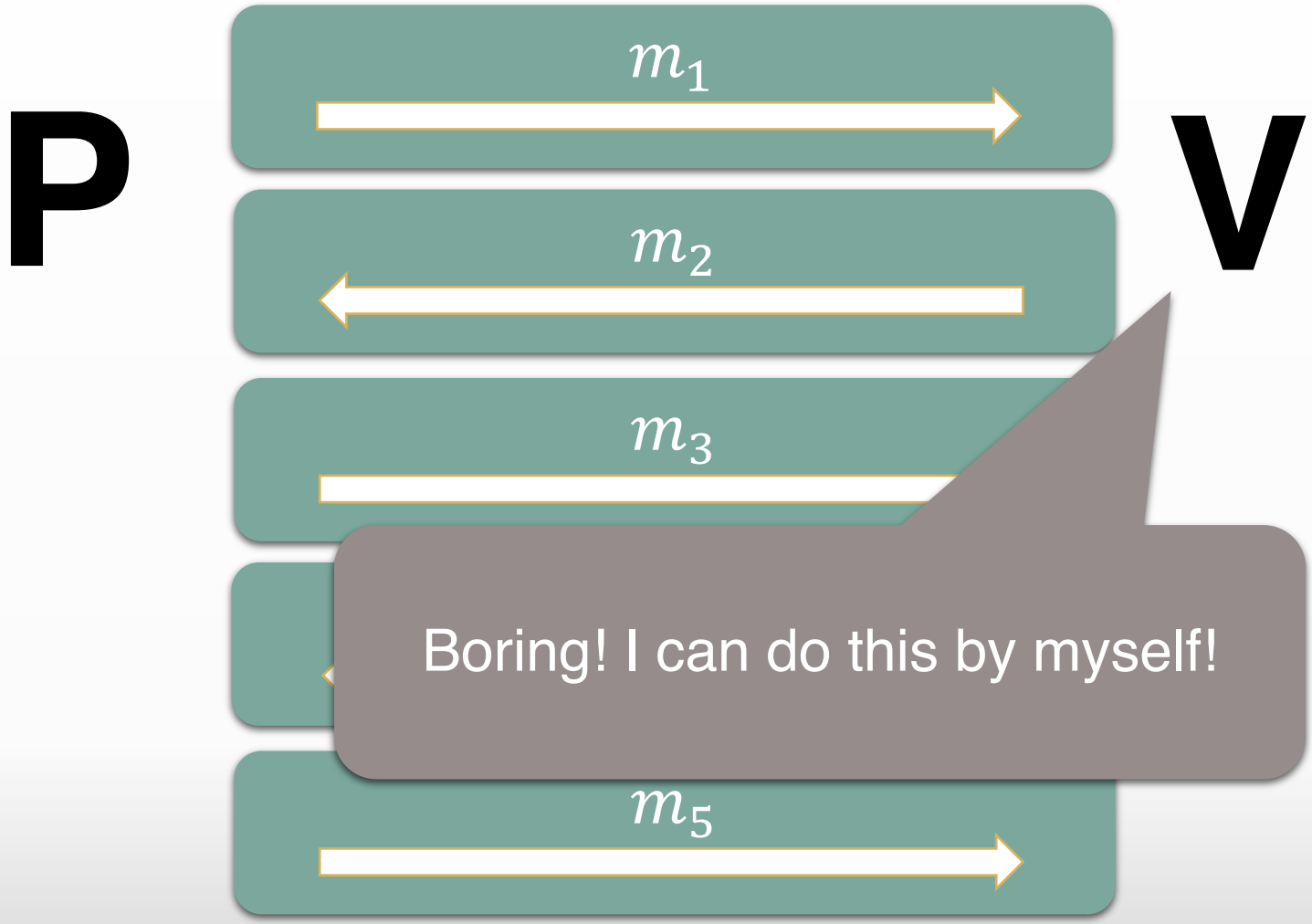
Intuitively:

- **V** is given information that $x \in \mathcal{L}$
- Modulo this, it could talked to itself

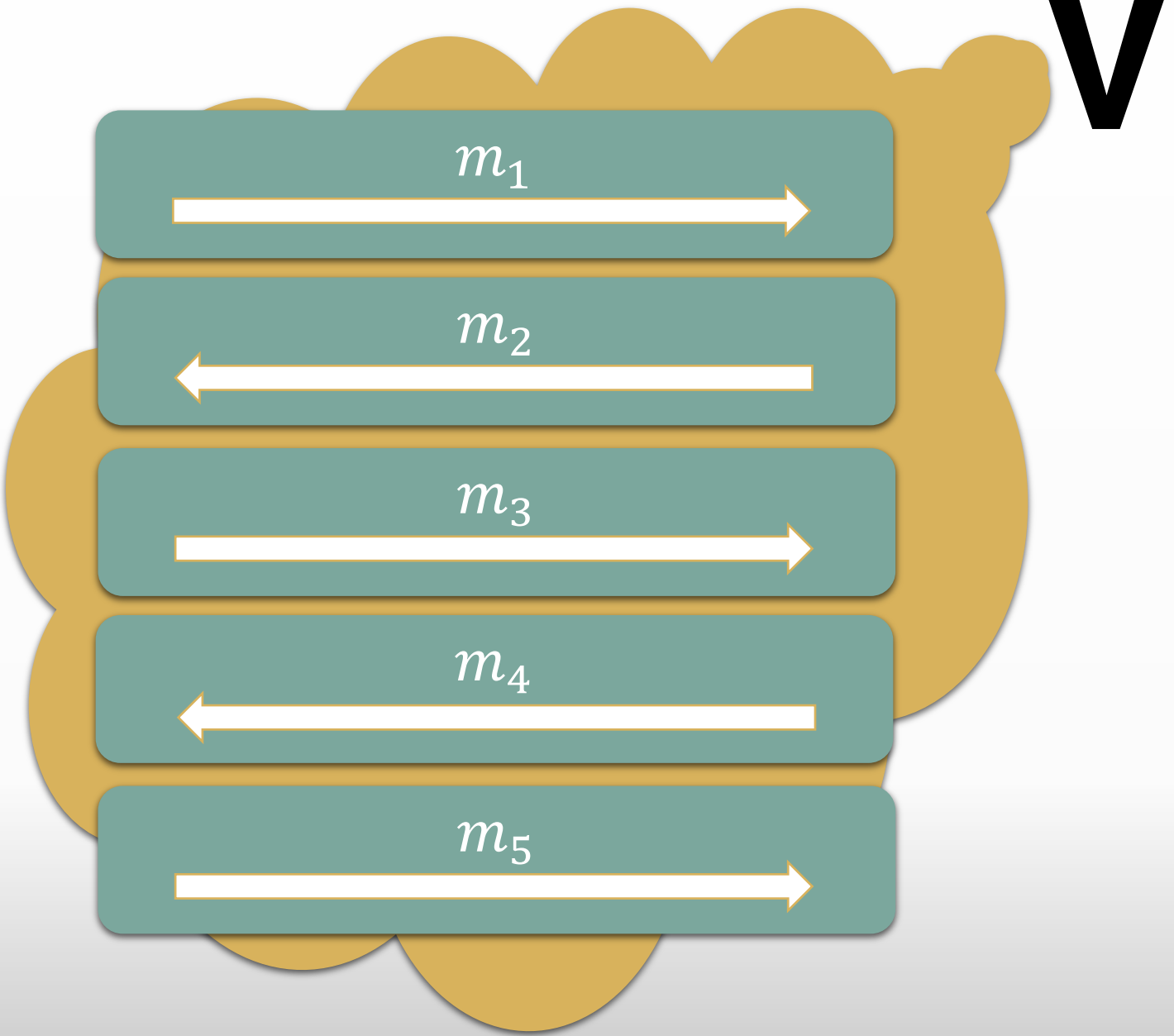
Technically

- $V(view) = V(simulation)$
- Whatever **V** could compute, he could compute even without talking with **P**

V does not need P



V does not need P



Zero Knowledge

V's view can be **simulated** in polynomial time

An interactive proof (P, V) for $\mathcal{L}$ is zero-knowledge if $\forall V$ there exists PPT S such that $\forall x \in \mathcal{L}$

$$S^V(x) \approx (P, V)(x)$$

$$\Pr_R [S^V(x) = (m_1, \dots, m_k)] \approx \Pr_R [(P, V)(x) = (m_1, \dots, m_k)]$$

Simulator can be probabilistic (probabilistic polynomial time)

S has **black-box** access to V if
It observers its inputs / outputs only (usually)

Zero Knowledge

V's view can be **simulated** in polynomial time

An interactive proof (P, V) for $\mathcal{L}$ is zero-knowledge if $\forall V$ there exists PPT S such that $\forall x \in \mathcal{L}$

$$S^V(x) \approx (P, V)(x)$$

$$\Pr_R [S^V(x) = (m_1, \ldots, m_k)] \approx \Pr[(P, V)(x) = (m_1, \ldots, m_k)]$$

It is **only** required simulator to exists!

no more, but **no less**

It observers its [illegible] only (usually)

Indistinguishability

$$S^V(x) \approx (P,V)(x)$$

$$SD(X,Y) = \frac{1}{2}\sum_x |\Pr[X = x] - \Pr[Y = x]|$$

$$SD\big(S(x),(P,V)(x)\big) = 0 \qquad \textbf{perfect ZK}$$

$$SD\big(S(x),(P,V)(x)\big) = negl(k) \qquad \textbf{statistical ZK}$$

Function $negl$ is negligible if for every polynomial $poly$ there exists k_0, such that for $k \geq k_0$: $negl(k) \leq \frac{1}{poly(k)}$

(e.g. 2^{-k})

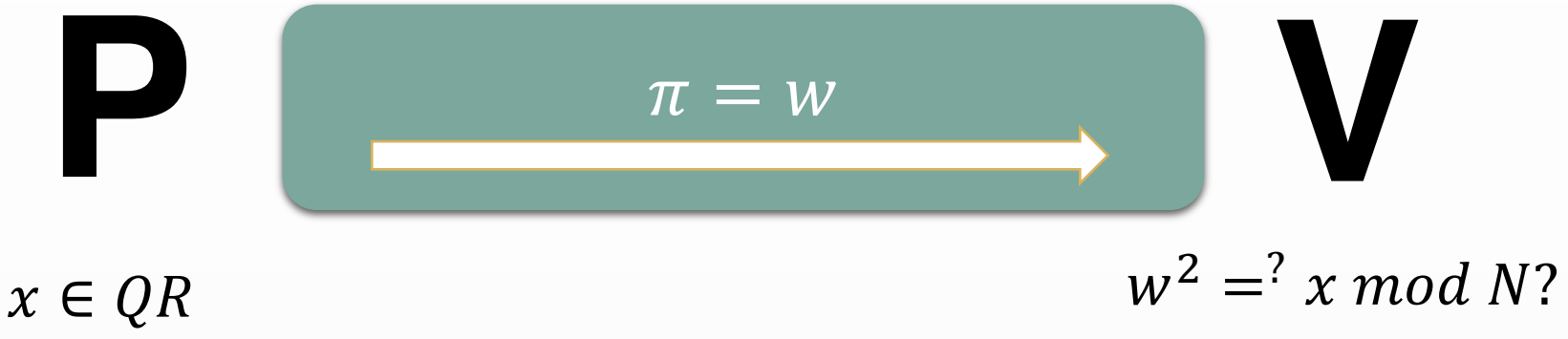
Different flavours of ZK

Interactive vs Non-interactive

	perfect	statistical	comp.
proof	X		
argument			

Quadratic residuosity again

$$QR_N = \{x \mid x \text{ is quadratic residure mod } N\}$$



Why it is not zero-knowledge?

1. For all $x \in QR_N$, $S(x)^2 = x \bmod N$
2. For all $x \notin QR_N$, $S(x)^2 \neq x \bmod N$
3. Since $QR \notin BPP$ then $\exists x \in QR_N$: $S(x)^2 \neq x \bmod N$

Quadratic residuosity – zk proof

$QR_N = \{x \mid x \text{ is quadratic residure mod } N\}$

P

$r \in_R \mathbb{Z}_N^*$

x, w

$w^2 = x \bmod N$

V

x

$y = r^2$

b

$b \in_R \{0,1\}$

$b = 0$: $z_0 = r$

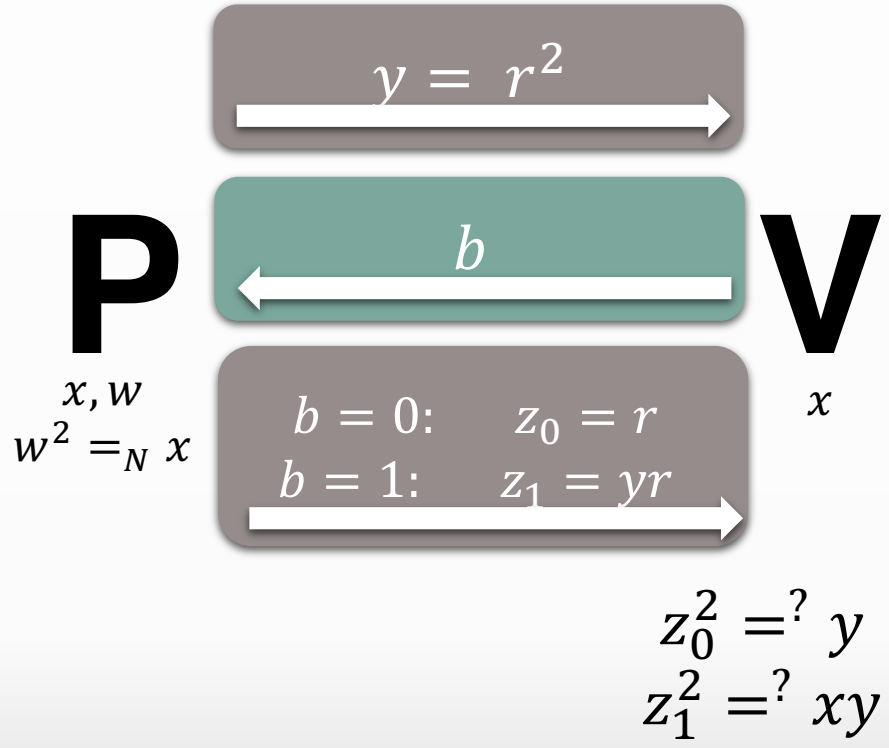
$b = 1$: $z_1 = yr$

$z_0^2 =^? y$

$z_1^2 =^? xy$

Why it is complete?

Quadratic residuosity – soundness



Soundness:

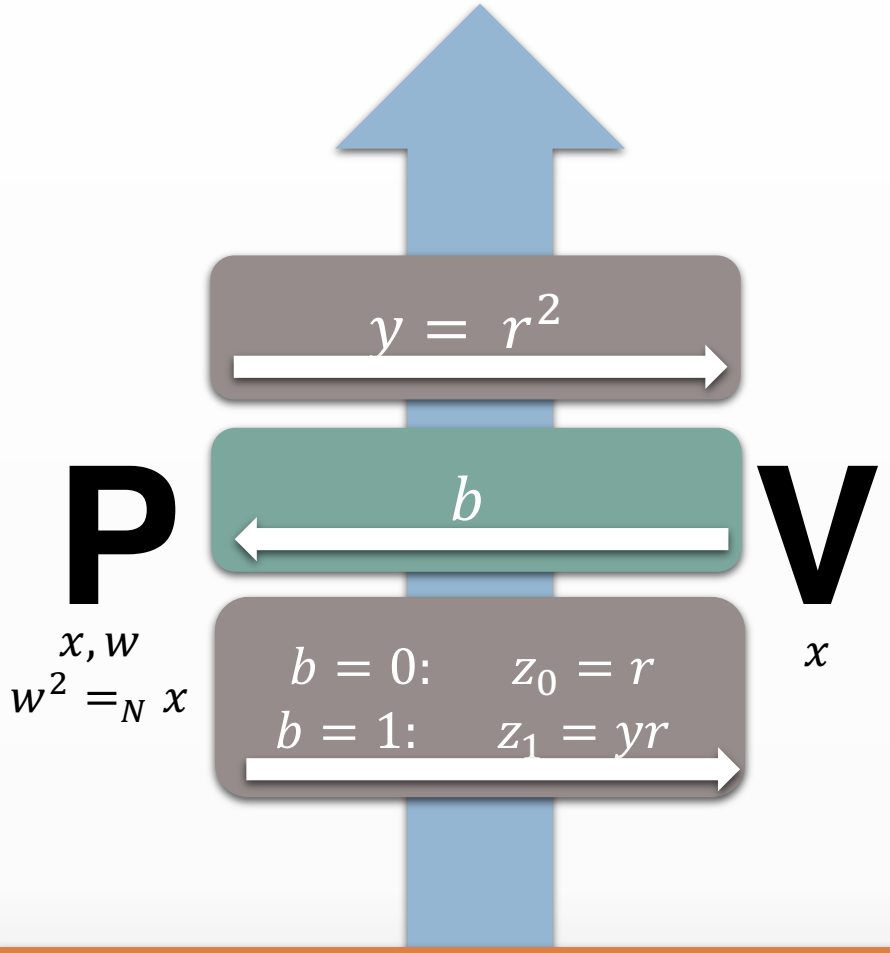
$x \in QR$ iff

$\exists y, y \in QR_N$ and $xy \in QR_N$

If $\Pr_b[(P^*, V) = 1] > \frac{1}{2}$

then both $z_0^2 = y$ and $z_1^2 = xy$

Quadratic residuosity – perfect zero-knowledge



The simulation:

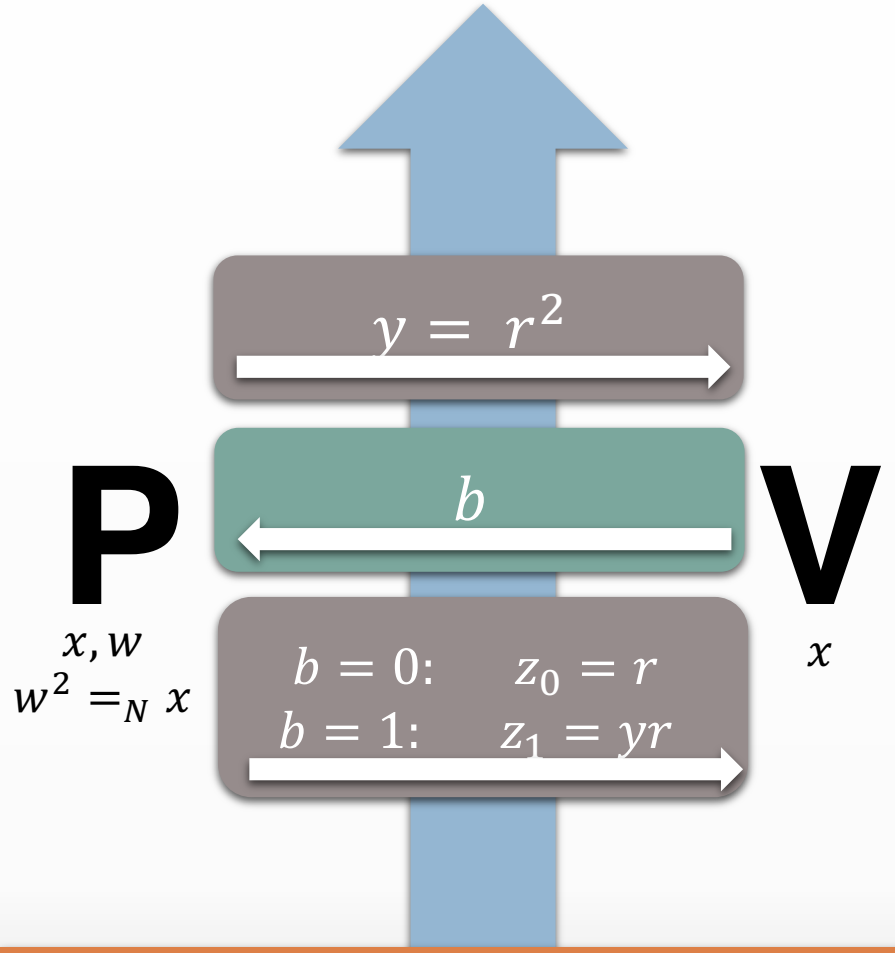
1. Sample $z \in_R \mathbb{Z}_N^*$
2. Sample $b \in_R \{0,1\}$
3. Set $y = \frac{z^2}{x^b}$
4. Output (y, b, z)

Random (y, b, z) such that $z^2 = x^b y$

$\approx$

Random (y, b, z) such that $z^2 = x^b y$

Quadratic residuosity – perfect honest verifier zero-knowledge



The simulation:

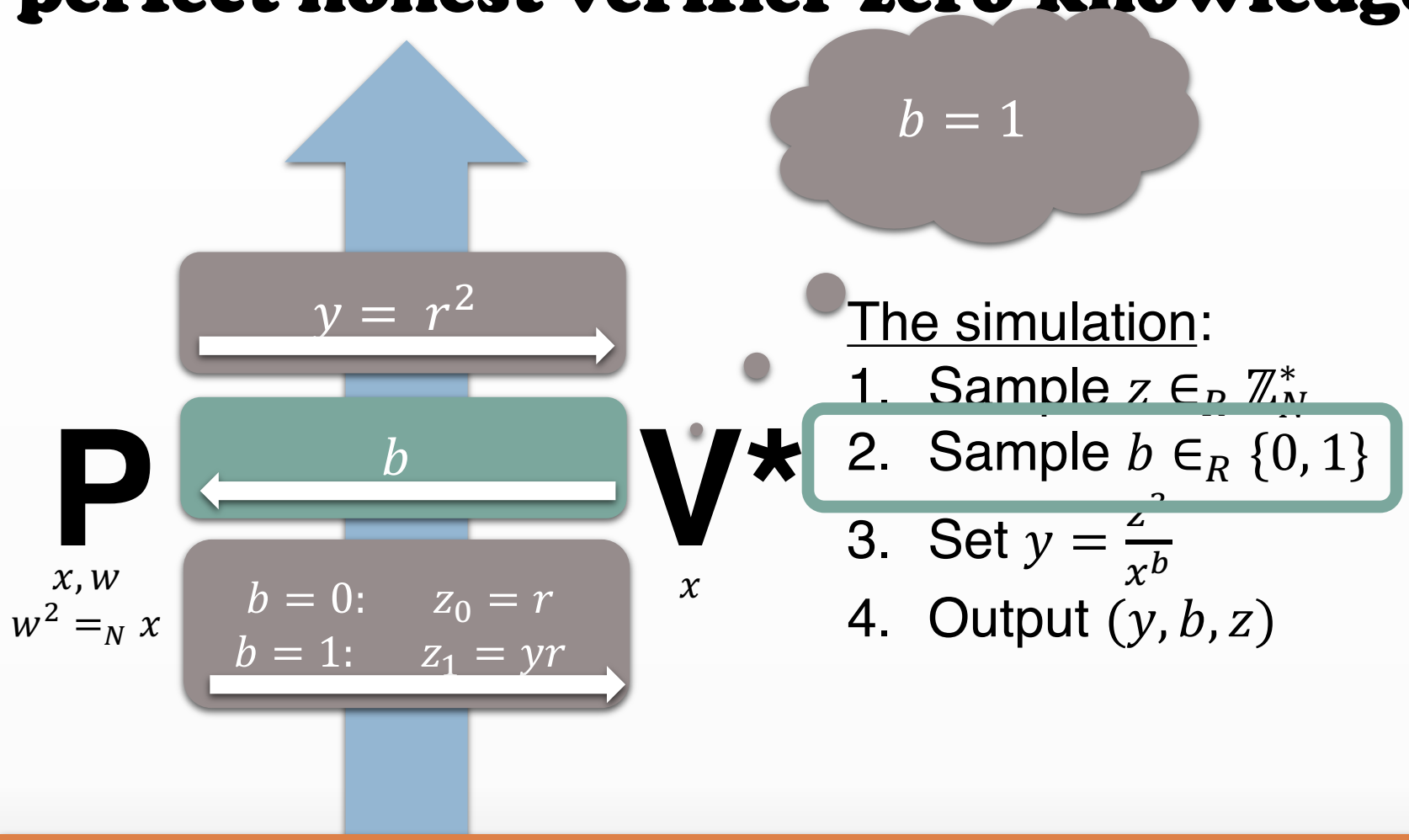
1. Sample $z \in_R \mathbb{Z}_N^*$
2. Sample $b \in_R \{0, 1\}$
3. Set $y = \frac{z^2}{x^b}$
4. Output (y, b, z)

Random (y, b, z) such that $z^2 = x^b y$

$\approx$

Random (y, b, z) such that $z^2 = x^b y$

Quadratic residuosity – perfect honest verifier zero-knowledge



The simulation:

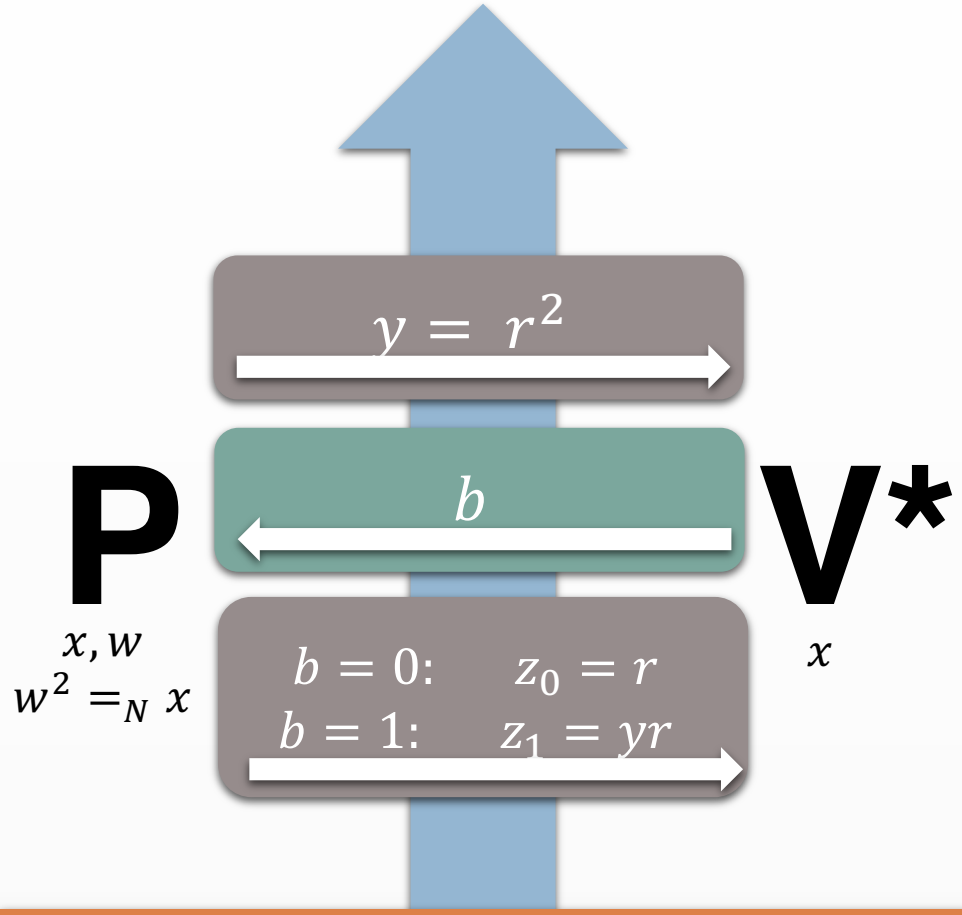
1. Sample $z \in_R \mathbb{Z}_N^*$
2. Sample $b \in_R \{0, 1\}$
3. Set $y = \frac{z^2}{x^b}$
4. Output (y, b, z)

Random (y, b, z) such that $z^2 = x^b y$

$\approx$

Random (y, b, z) such that $z^2 = x^b y$

Quadratic residuosity – perfect zero-knowledge



The simulation:

1. Sample $z \in_R \mathbb{Z}_N^*$
2. Sample $b \in_R \{0, 1\}$
3. Set $y = \frac{z^2}{x^b}$
4. If $V^*(y) = b$ output (y, b, z), else **repeat**

Exp. number of iterations: 2

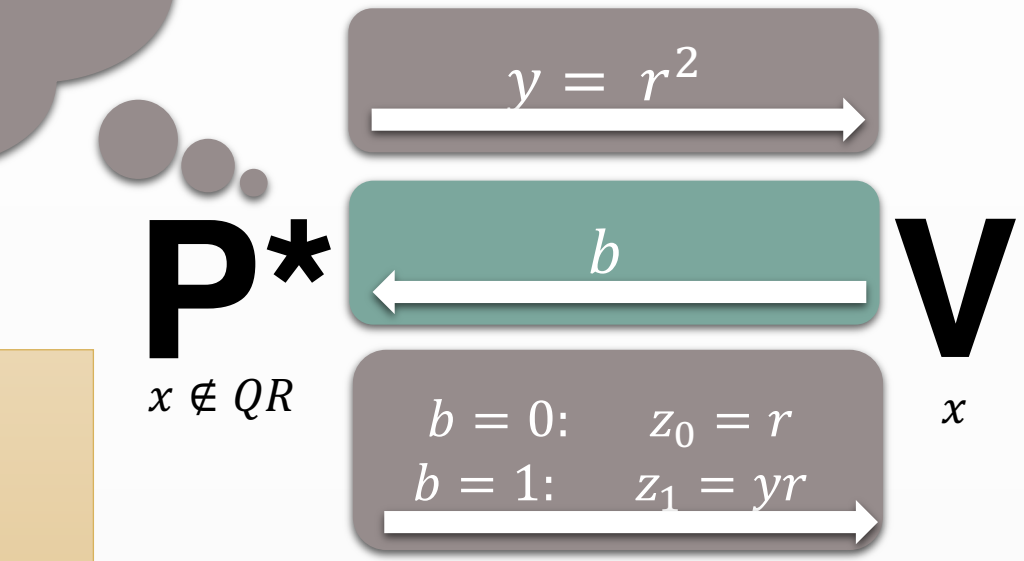
Random (y, b, z) such that $z^2 = x^b y$ and $V^*(y) = b$

$\approx$

Random (y, b, z) such that $z^2 = x^b y$ and $V^*(y) = b$

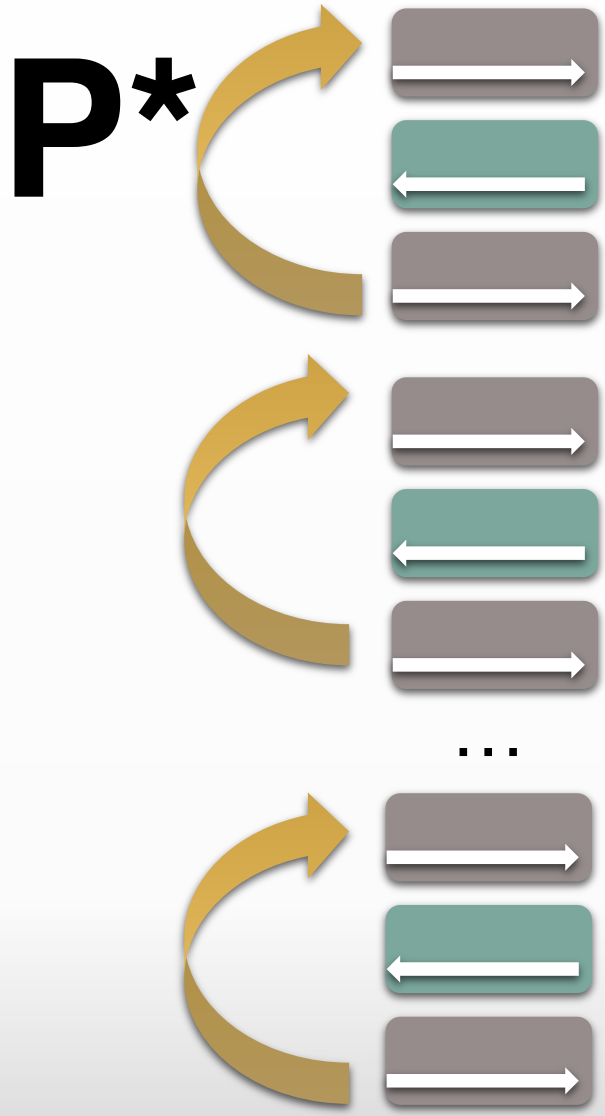
Cheating prover

If I guess b correctly I can pick y and z to convince V



P^* can cheat V with probability $\frac{1}{2}$

Amplifying soundness



Sequential composition

ACCEPT if all repetitions accept

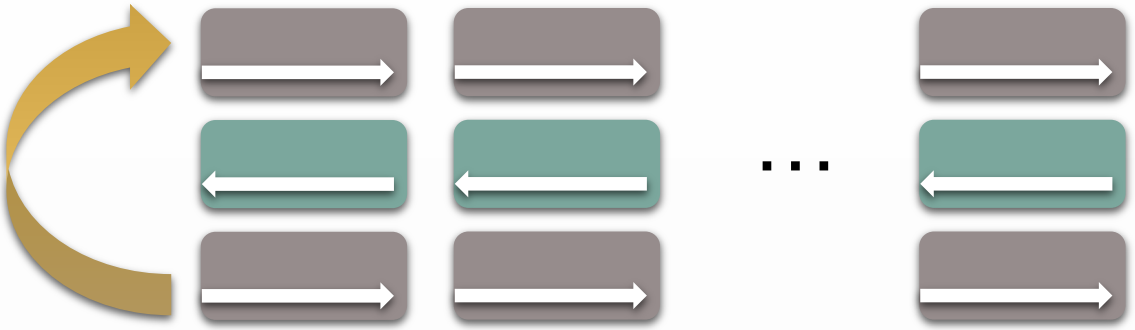
n iterations gives soundness $\leq 1/2^n$

How to simulate?

Simulator rewinds each iteration separately

$\mathbb{E}[time(S)] = 2\ time(V^*)$

Amplifying soundness



Parallel composition

ACCEPT if all repetitions accept

n iterations gives soundness $\leq 1/2^n$

How to simulate?

S has to rewind all blocks at the same time

$\mathbb{E}[time(S)] = 2^k time(V^*)$

Impossible in a black-box model

Possible with some relaxations

Auxiliary input

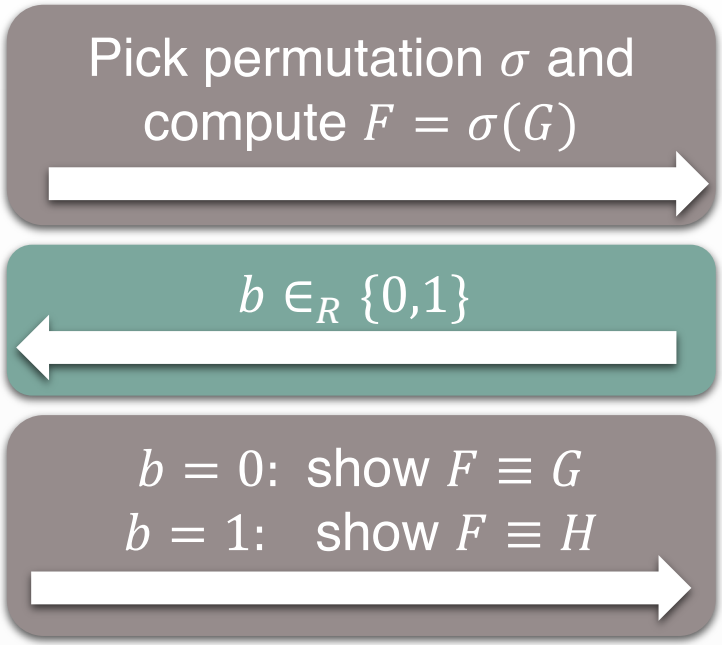
ZK proof for GI

$$GI_G = \{H \mid H \text{ is isomorphic to } G\}$$

P

$H \in GI_G$

$f: f(G) = H$



V

H

The protocol is **zero-knowledge** (food for thought)

Fact: Let $G = (V, E)$ and V public
Then $f(G)$ is a random element of set of all graphs isomorphic to G

ZK proof for $\overline{GI}$

$$\overline{GI}_G = \{H \mid H \textit{ is } \textbf{not} \textit{ isomorphic to } G\}$$

P

$G, H \in GI_G$
(can show it for any relabeling of G and H)

Take (G, H) and randomly relabel it in a random order (F_1, F_2)

Decide which F_1, F_2 is isomorphic to G

V

H

The protocol is **not zero-knowledge** if we allow V to take some auxiliary input

V may use P to learn whether any of F_1, F_2 is isomorphic with G

ZK proof for $\overline{GI}$

$$\overline{GI}_G = \{H \mid H \text{ is } \textbf{not} \text{ isomorphic to } G\}$$

P

$G, H \in GI_G$
(can show it for any relabeling of G and H)

Take (G, H) and randomly relabel it in a random order (F_0, F_1)

Show in ZK that $(G, H) \equiv (F_1, F_2)$

Decide which F_0, F_1 is isomorphic to G

V

H

Soundness (intuition)

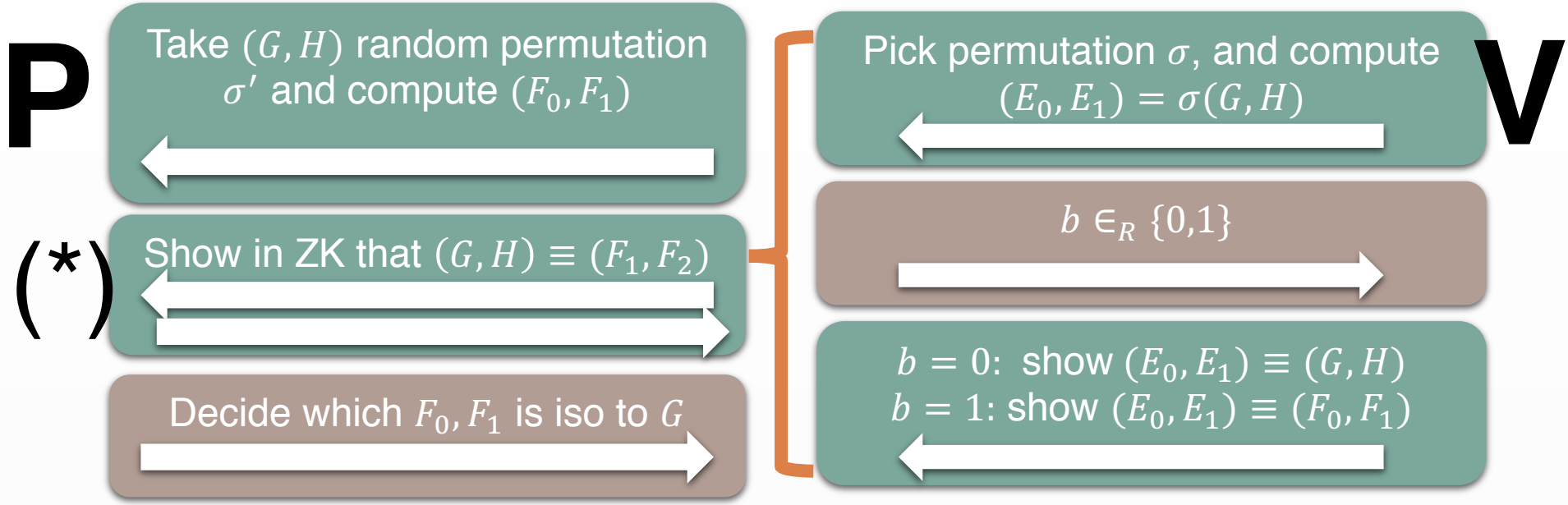
P^* can cheat V with probability $\frac{1}{2}$ (random guess)

Zero knowledge (intuition)

P only shows which F_b isomorphic to G. This is already known to V

ZK proof for $\overline{GI}$

$$\overline{GI}_G = \{H \mid H \not\equiv G\}$$



Zero knowledge

1. Simulator S picks randomly bit b
   - If $b = 0$: S learns permutation σ
   - If $b = 1$, S learns permutation $\sigma \circ \sigma'$
2. Simulator S rewinds the verifier and pick b again
3. When picked bits are different, S learns both σ and $\sigma \circ \sigma'$, thus can compute σ' and decide which F_0, F_1 is isomorphic to G

Auxiliary input

DEFINITION: Interactive proof (P, V) is **zero-knowledge wrt auxiliary input** if for every PPT V^* $\forall x \in \mathcal{L}, \forall z$

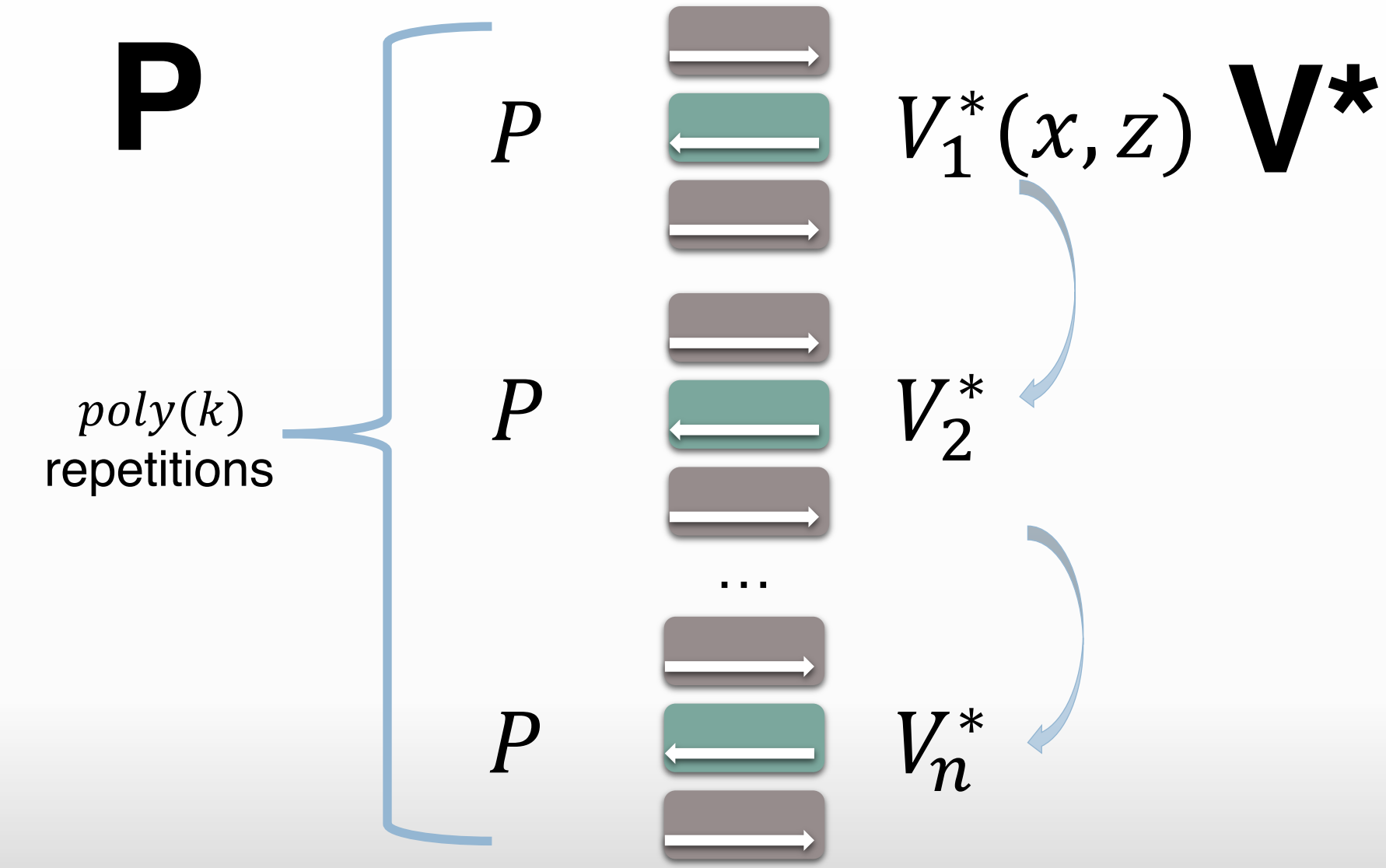
$$S^{V^*}(x, z) \approx (P, V(z))(x)$$

- Catches context in which the protocol is executed
- Useful if V may have some a'priori information about w

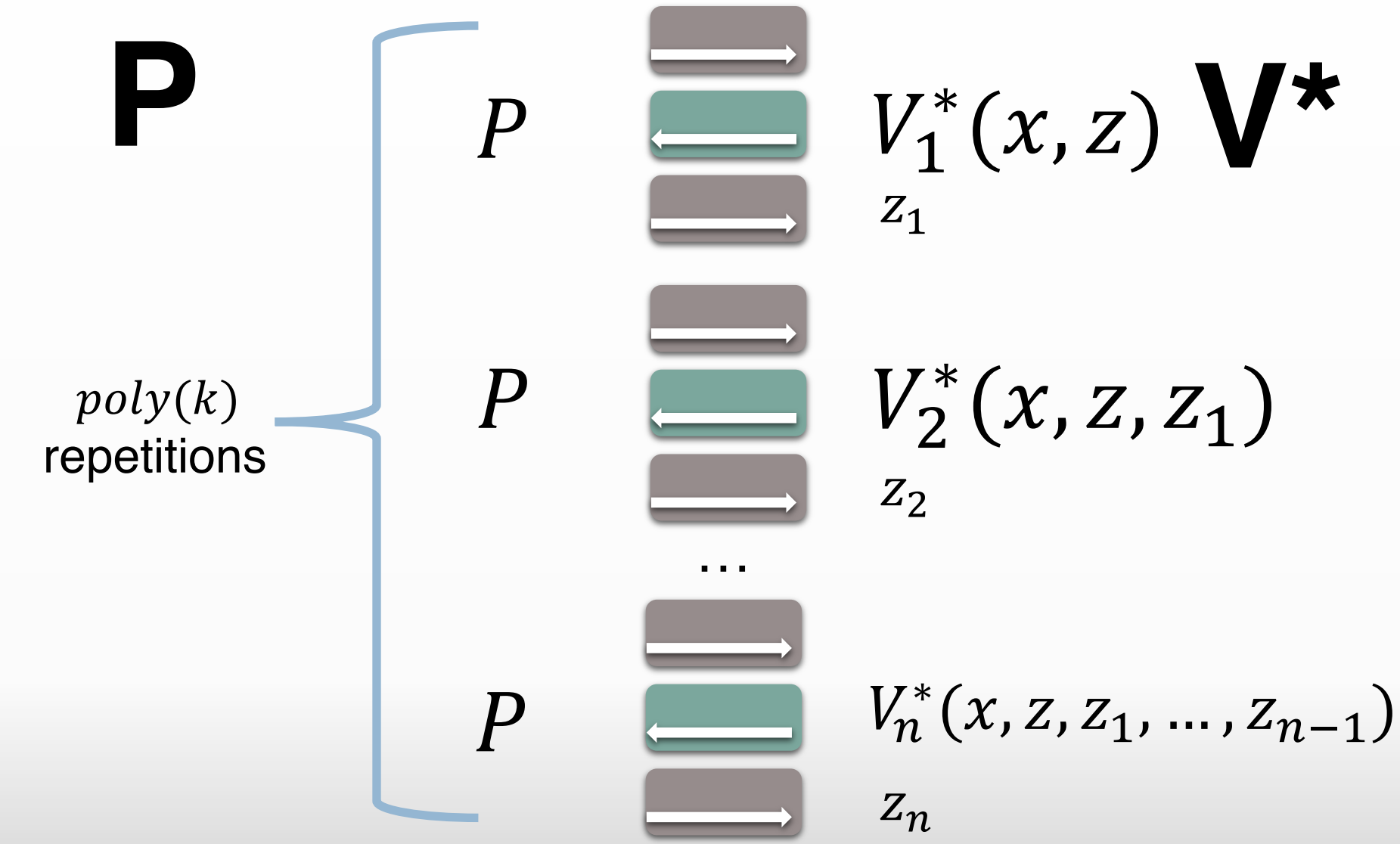
Simulator also gets the auxiliary input

Crucial for composition

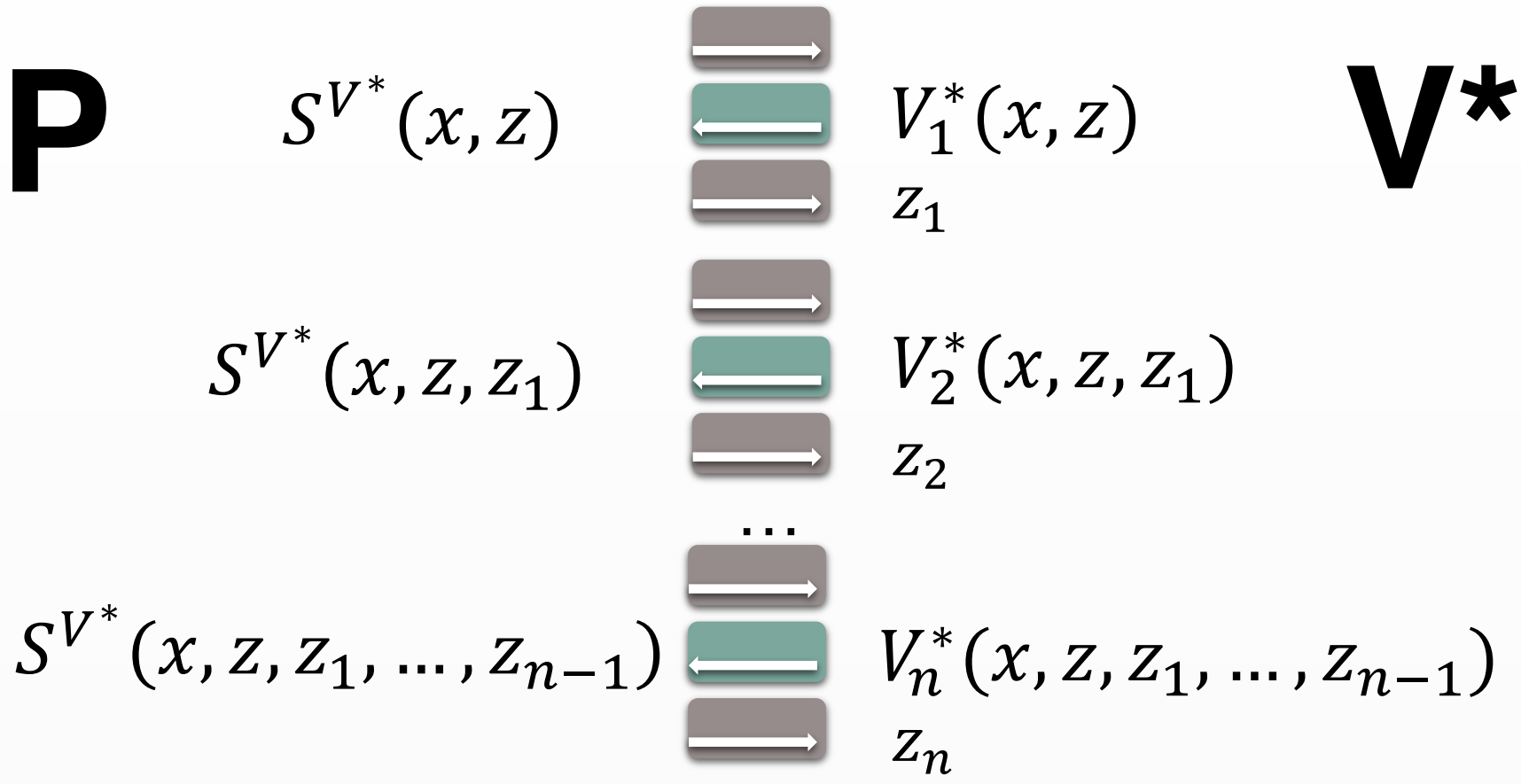
Auxiliary input and composition



Auxiliary input and composition



Auxiliary input and composition



DEFINITION: Interactive proof (P,V) is **zero-knowledge wrt auxiliary input** if for every PPT V^* $\forall x \in \mathcal{L}, \forall z$

$$S^{V^*}(x,z) \approx (P,V(z))(x)$$

SAT and perfect ZK

$$SAT = \{\phi \mid \phi \; is \; satisfiable \; boolean \; formula\}$$

$$SAT = \{\phi(w_1, \dots, w_n) \mid \exists \, w \in \{0,1\}^n \, , \phi(w) = 1\}$$

If there is PZK for SAT, then polynomial hierarchy collapses

Possible relaxation:
statistical / computational ZK computational soundness

If there is SZK for SAT, then polynomial hierarchy collapses

Different flavours of ZK

Interactive vs Non-interactive

	perfect	statistical	comp.
proof			X
argument			

Computational ZK

Computational ZK

$$\forall PPT\ V^* \exists PPT\ S\ \forall x \in \mathcal{L} \forall z$$
$$S(x,z) \approx_c (P, V^*(z))(x)$$

X and Y are computationally ϵ-close if for every PPT algorithm D

$$|\Pr[D(X) = 1] - \Pr[D(Y) = 1]| \leq \epsilon$$

That is, no efficient algorithm can distinguish X and Y much better than randomly

Computational ZK

$$PZK \subset SZK \subset CZK$$

THEOREM: Suppose one-way functions exist, then $\mathbb{NP} \subset CZK$

DEFINITION: $f: A \to B$ is (t, ϵ)-one-way if for all adversaries $\mathcal{A}$ that works in time t

$$\Pr_X[A \; inverts \; f(X)] \leq \epsilon$$

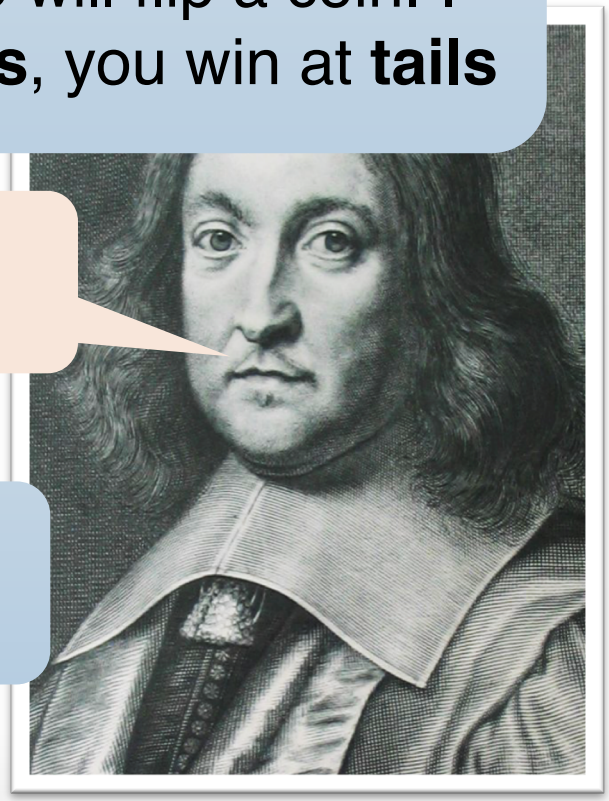
Commitment schemes

Coin-flipping over phone

Let's make a lottery over phone. We will flip a coin. I win at **heads**, you win at **tails**

Ok!
$b_1 = 1$

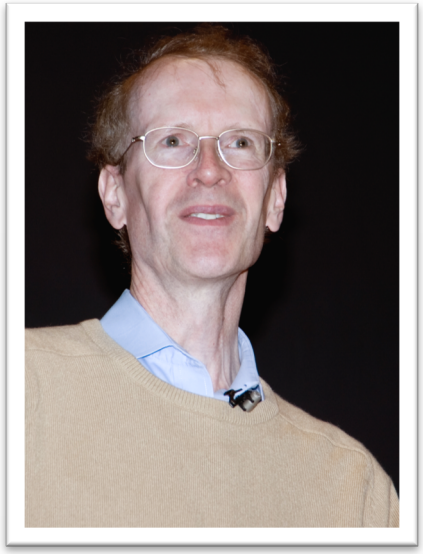
$b_0 = 1$
I won!



Andrew picks bit b_0, Pierre picks bit b_1 and the result is $b_0 + b_1 mod\ 2$
head = 0, **tails** = 1

Security

Goal:
Have a protocol π that ends in bit b that is uniformly distributed over $\{0, 1\}$

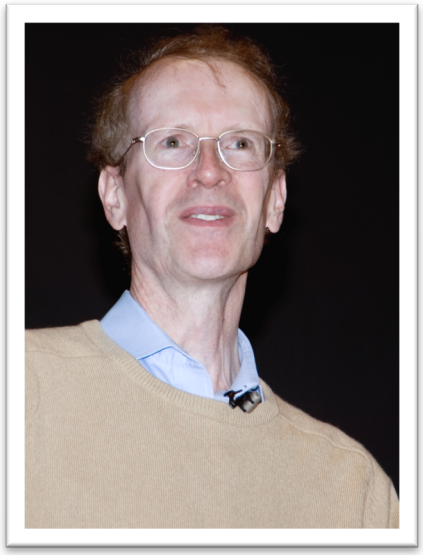


Andrew: Even if Pierre is cheating, π will produce an uniform output

Pierre: Even if Andrew is cheating, π will produce an uniform output

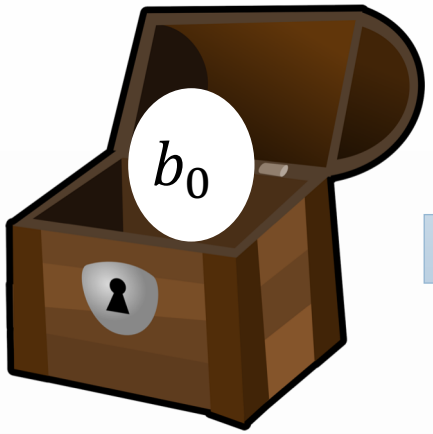
Security

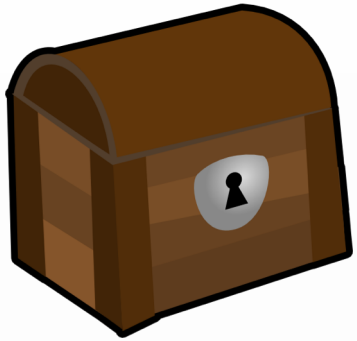
Goal:
Have a protocol π that ends in bit b that is uniformly distributed over $\{0, 1\}$



Who sends the bit first loses?

Commitment scheme





C

R

Committer cannot change the content after it was sent

Receiver cannot check the content before they get the key

Commitment scheme

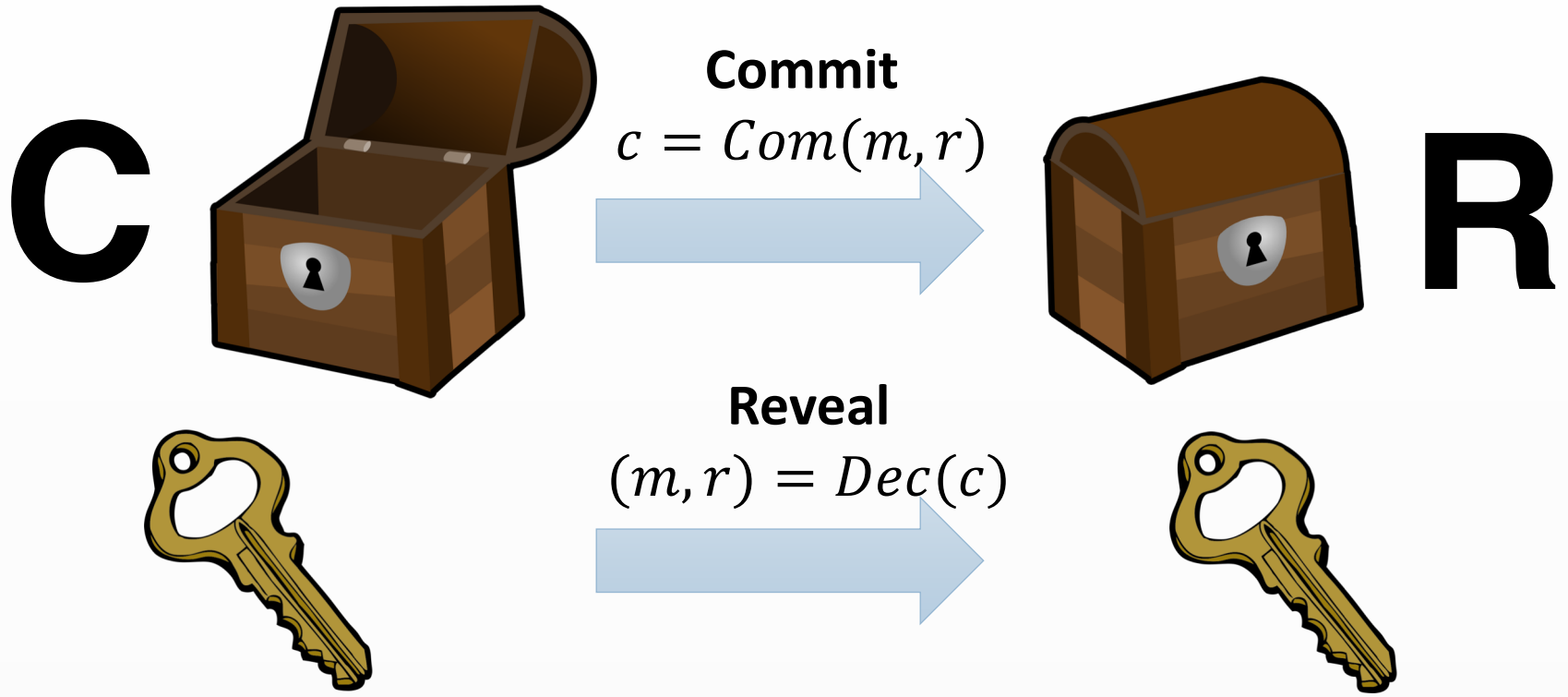
BINDING

Committer cannot change the content after it was sent

HIDING

Receiver cannot check the content before they get the key

Commitment scheme



Receiver given commitment c and its opening m', r' checks whether $Com(m', r') = c$

Perfectly-hiding commitment schemes

Perfectly hiding commitment scheme (Com, Dec) is

- **Perfectly hiding** $\forall m_1, m_2$

$$\Pr_{r}[Com(m_1, r) = c] = \Pr_{r}[Com(m_2, r) = c]$$

- **Computationally binding** $\forall PPT C^* \; \forall m_1 \neq m_2$

$$\Pr[C^* \; wins \; the \; binding \; game] \leq negl(n)$$

Binding game:

C^* wins the binding game if it generates c, m_1, r_1, m_2, r_2 such that

$$c = Com(m_1, r_1) = Com(m_2, r_2) \text{ for } m_1 \neq m_2$$

Pedersen commitment scheme

Discrete logarithm problem:
Let $\mathbb{G}$ be a finite group and g a generator of a subgroup in $\mathbb{G}$, we say $\mathcal{A}$ solves discrete logarithm problem if $\mathcal{A}(\mathbb{G}, g, g^x) = x$

Discrete logarithm assumption
Let g be a generator of a subgroup of $\mathbb{Z}_p^*$ then discrete logarithm is hard

Pedersen commitment

Let $g, h = g^x$ be $\mathbb{Z}_p^*$ elements such that no one knows x

$$Com(m, r) = g^m h^r$$

Pedersen commitment- binding

Pedersen commitment

Let $g, h = g^x$ be $\mathbb{Z}_p^*$ elements such that no one knows x then

$$Com(m, r) = g^m h^r$$

Computational binding:

Assume $\mathcal{A}$ could open c to 2 different (m, r) and (m', r') then

$$g^m h^r = g^{m'} h^{r'}$$

$$g^{m-m'} = h^{r'-r}$$

$$g^{\frac{m-m'}{r-r'}} = h$$

$$x = \frac{m - m'}{r - r'}$$

Pedersen commitment- hiding

Pedersen commitment

Let $g, h = g^x$ be $\mathbb{Z}_p^*$ elements such that no one knows x then

$$Com(m, r) = g^m h^r$$

Perfect hiding:

Assume $\mathcal{A}$ given c can hides m or m', but for all m, r, m' there is r' such that

$$Com(m, r) = c = Com(m', r')$$

Statistically-binding commitment schemes

Statistically binding commitment scheme (Com, Dec) is

- **Computational hiding** $\forall PPT\ R^*\ \forall m_1, m_2$

$$Com(m_1) \approx_c Com(m_2)$$

- **Statistical binding** $\forall C^*\ \forall m_1 \neq m_2$

$$\Pr[C^*\ wins\ the\ binding\ game] \leq negl(n)$$

Binding game:

C^* wins the binding game if it generates c, m_1, r_1, m_2, r_2 such that

$$c = Com(m_1, r_1) = Com(m_2, r_2) \text{ for } m_1 \neq m_2$$

Statistically binding commitment schemes

$PRG: \{0,1\}^n \rightarrow \{0,1\}^{3n}$

Given random input of length n outputs pseudorandom output of length $3n$

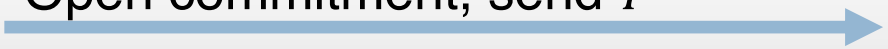
pseudorandom = no efficient machine can tell it from a random string

C

Selects a random $X \in \{0,1\}^{3n}$

Selects a random $Z \in \{0,1\}^n$
If $b = 0$ sends $Y = PRG(Z) \oplus X$
If $b = 1$ sends $Y = PRG(Z)$

Open commitment, send Y



R

Check
$Y = PRG(Z) \oplus X, b \leftarrow 0$
$Y = PRG(Z), b \leftarrow 1$

Note: each public key cryptosystem is perfectly binding, comp. hiding commitment

Statistical binding

Proof intuitions

To be able to cheat committer has to find Z and Z' such that

$$PRG(Z) \oplus X = Y = PRG(Z')$$

Thus $PRG(Z) \oplus PRG(Z') = X$

How many X's has property that there **exist** Z and Z' that $PRG(Z) \oplus PRG(Z') = X$?

By counting argument, at most $(2^n)^2 = 2^{2n}$

There the probability that random $X \in \{0,1\}^{3n}$ has this property is

$$\frac{2^{2n}}{2^{3n}} = 2^{-n} \text{ (negligible)}$$

Perfectly binding and perfectly hiding commitment?

Impossible ☹

Perfectly hiding:

For each m, m' they are r, r' such that

$Com(m, r) = Com(m', r')$

Otherwise adversary could find m by exhaustive search over all possible m and r and break hiding

Perfectly binding:

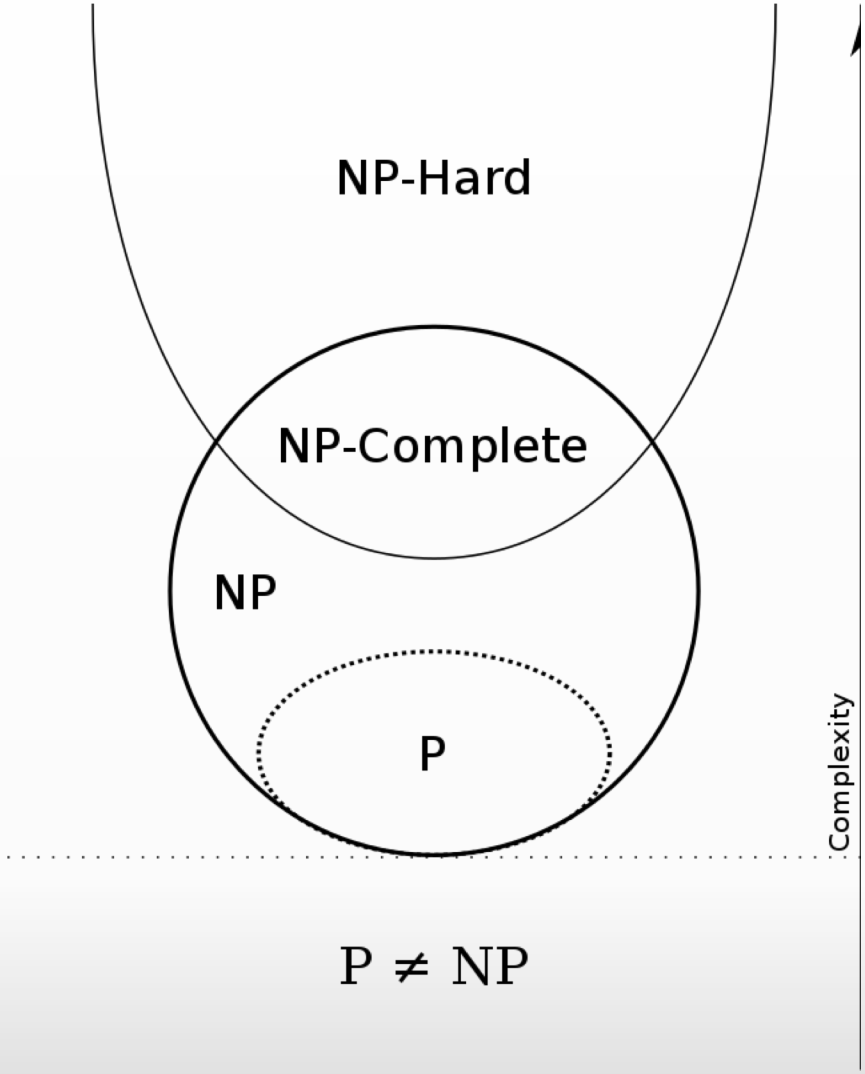
For each m, m' and all r, r'

$$Com(m, r) \neq Com(m', r')$$

Otherwise adversary could find m, m' by exhaustive search over all possible m, m' and r, r' and break binding

$$\mathbb{NP} \subset CZK$$

NP-complete problems



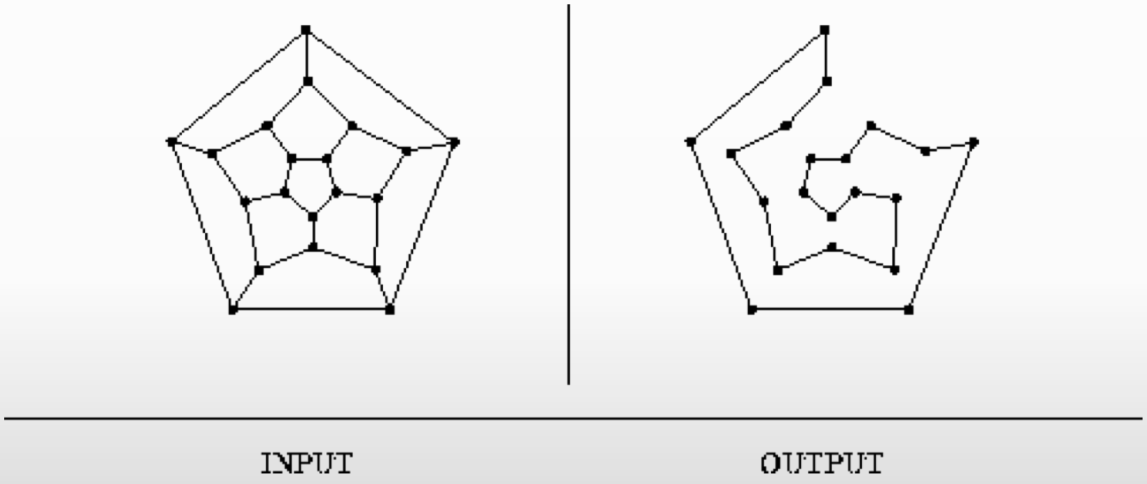
Recall
If we can solve at least **one** problem from that is $\mathbb{NP}$-complete we can solve all of $\mathbb{NP}$ problems

Graph Hamiltonicity

$$HAM = \{G \mid G\ has\ a\ Hamiltonian\ cycle\}$$

Hamiltonian cycle – a cycle that visit each node exactly once

Finding a Hamiltonian cycle in a graph is $\mathbb{NP}$-complete
(i.e. is at least as hard as any other problem in the $\mathbb{NP}$ class)



Graph Hamiltonicity

$$HAM = \{G \mid G \text{ has a Hamilton cycle}\}$$

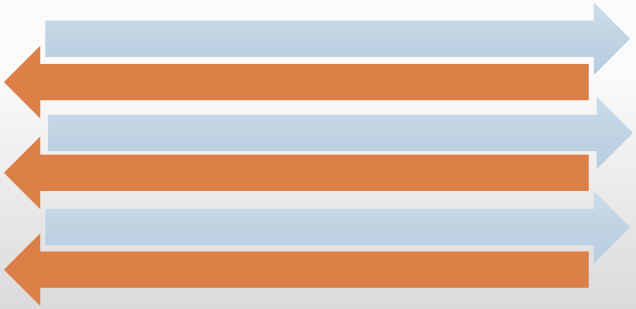
Finding a Hamilton cycle in a graph is $\mathbb{NP}$-complete
(i.e. is at least as hard as any other problem in the $\mathbb{NP}$ class)

Every $\mathcal{L} \in \mathbb{NP}$ is poly-time reducible to HAM
$\exists$ poly-time computable function f such that for all x
$x \in \mathcal{L} \iff f(x) \in HAM$

To prove $\mathcal{L} \in CZK$ sufficient to show $HAM \in CZK$

P

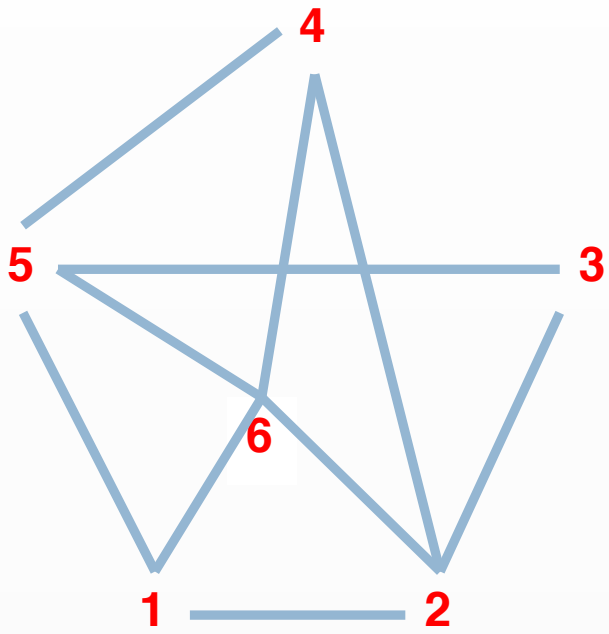
w for $x \in \mathcal{L} \leftrightarrow$
$g(w)$ for $f(x) \in$
HAM



$x \in \mathcal{L}$
$\leftrightarrow f(x)$
$\in HAM$

V

Adjacency matrix



	1	2	3	4	5	6
1	0	1	0	0	1	1
2	1	0	1	1	0	1
3	0	1	0	0	1	0
4	0	1	0	0	1	1
5	1	0	1	1	0	1
6	1	1	0	1	1	0

ZK proof for HAM

P

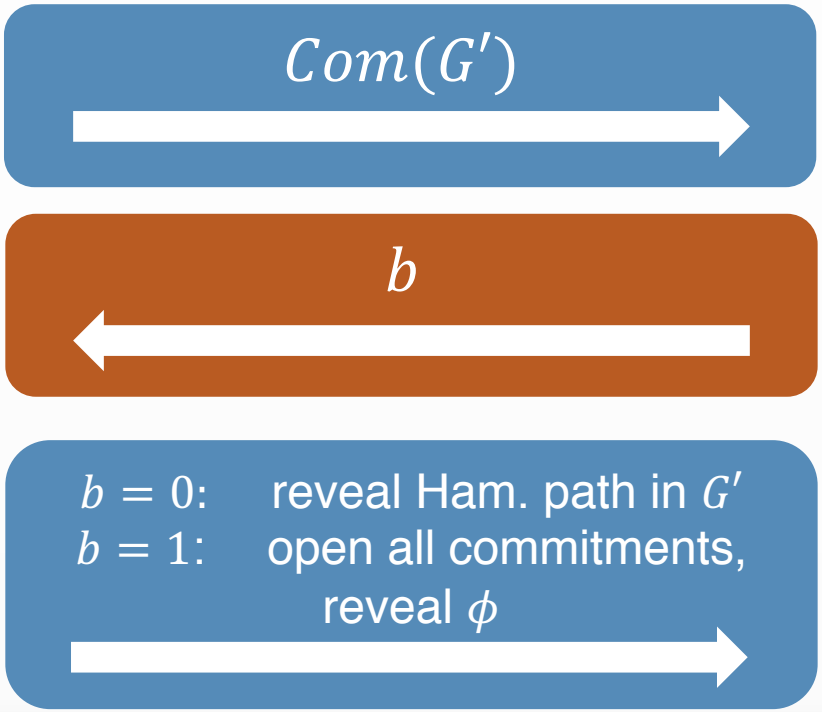
$G = (V, E)$ – Hamiltonian graph,
w – Hamiltonian path in G

$G = (V, E)$ – Hamiltonian graph

V

Pick permutation ϕ and compute
$G' = \phi(G)$

Commit to each entry of the adjacency matrix.



Pick random bit b

$b = 0$ – reveal cycle

$\phi(1) = 6$

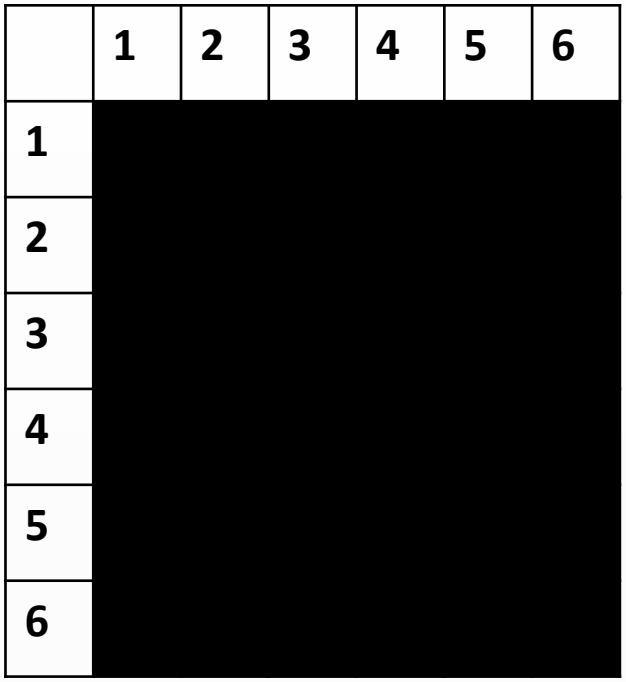
$\phi(2) = 3$

$\phi(3) = 4$

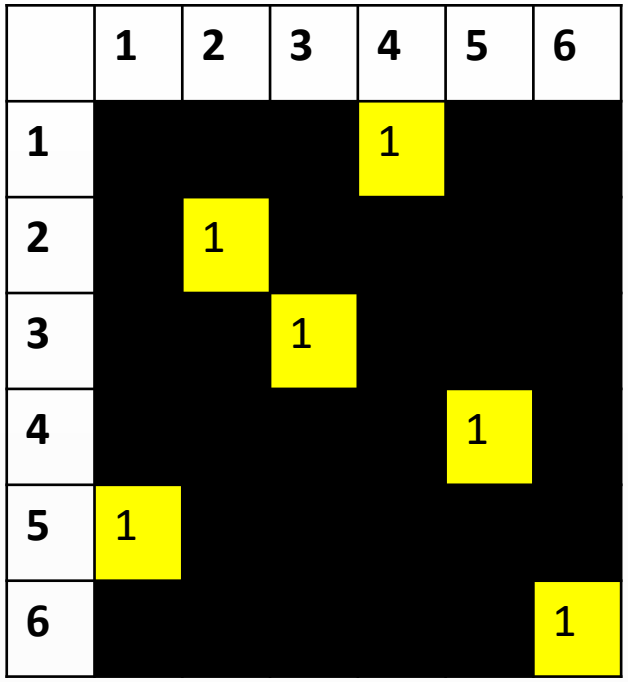
$\phi(4) = 2$

$\phi(5) = 5$

$\phi(6) = 1$



$c = Com(\phi(G))$



$u \in Dec(c)$

Decommitment is valid
u is a Hamiltonian cycle

$b = 1$ – reveal permutation

$\phi(1) = 6$

$\phi(2) = 3$

$\phi(3) = 4$

$\phi(4) = 2$

$\phi(5) = 5$

$\phi(6) = 1$

	1	**2**	**3**	**4**	**5**	**6**
1	0	1	0	0	1	1
2	1	0	1	1	0	1
3	0	1	0	0	1	0
4	0	1	0	0	1	1
5	1	0	1	1	0	1
6	1	1	0	1	1	0

G

	1	**2**	**3**	**4**	**5**	**6**
1	1	1	0	1	1	0
2	0	1	0	0	1	1
3	1	0	1	1	0	1
4	0	1	0	0	1	0
5	1	0	1	1	0	1
6	0	1	0	0	1	1

$G \approx G'$

Decommitment is valid
ϕ is a permutation

HAM soundness

If $\Pr_{b}[(P^*, V)\ accepts\ G] > \frac{1}{2}$ then both

- u is a Hamiltonian cycle in G'
- $G' = \phi(G)$

$\phi^{-1}(u)$ is a Hamiltonian cycle in G

HAM zero-knowledge

Simulator $S^{V^*}(G)$

1. Sample $b \in \{0,1\}$
   - If $b = 0$ pick randomly a graph G' with a Hamiltonian cycle u,
   - If $b = 1$ pick a random permutation ϕ and compute $G' = \phi(G)$
   - Compute $\mathrm{c} = \mathrm{Com}(G')$ and send it to V^*
2. If $V^*(c) = b$
   - $b = 0$: output (c, b, u)
   - $b = 1$: output $(c, b, (\phi, G'))$
3. Else **repeat**

Importance of commitment

Since Com is **computationally hiding** then S^{V^*} runs in polynomial time

Since Com is hiding and V^* is PPT

$$\Pr_{c,b}\left[V^*\left(Com\left(G'(b)\right)\right) = b\right] \approx 1/2$$

Otherwise V^* can distinguish $Com(G'(0))$ and $Com(G'(1))$

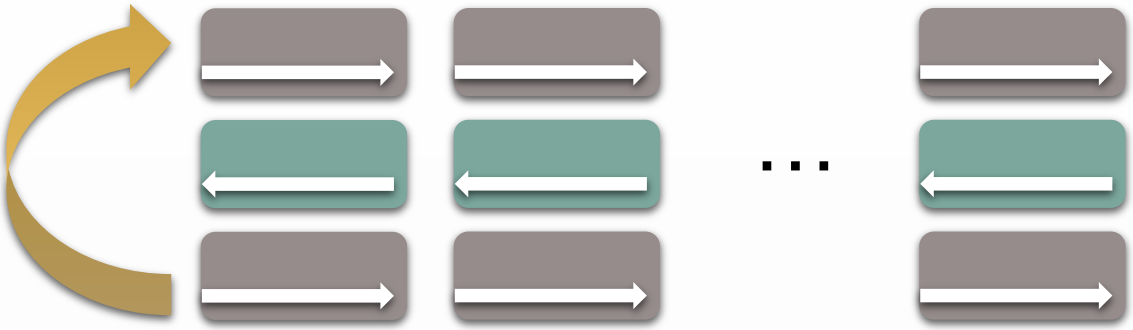
Since Com is **statistically binding** then no P^* can open the commitment in two ways

Different flavours of ZK

Interactive vs Non-interactive

	perfect	statistical	comp.
proof			
argument	X		

Amplifying soundness



Parallel composition

ACCEPT if all repetitions accept

n iterations gives soundness $\leq 1/2^n$

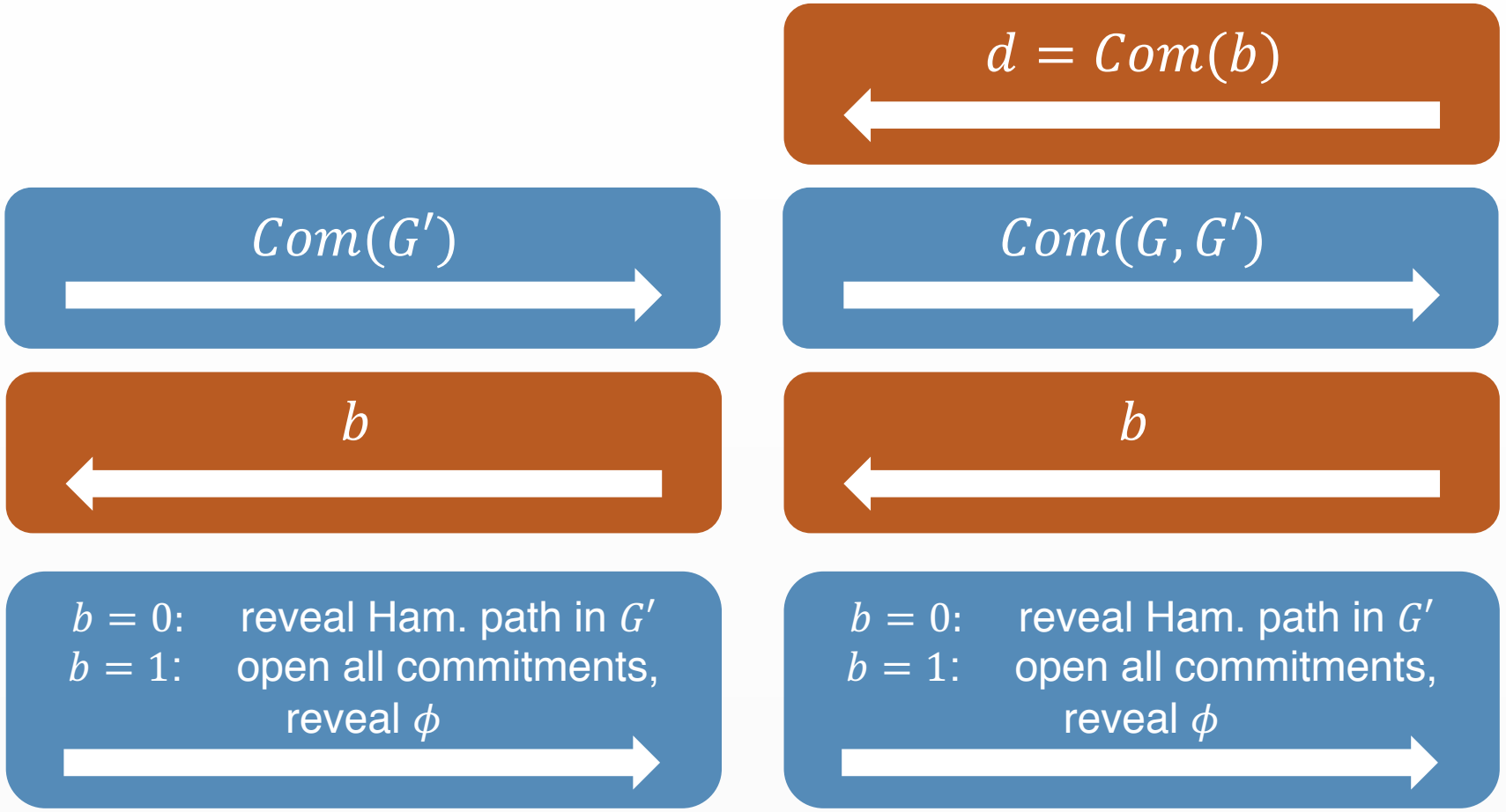
How to simulate?

S has to rewind all blocks at the same time

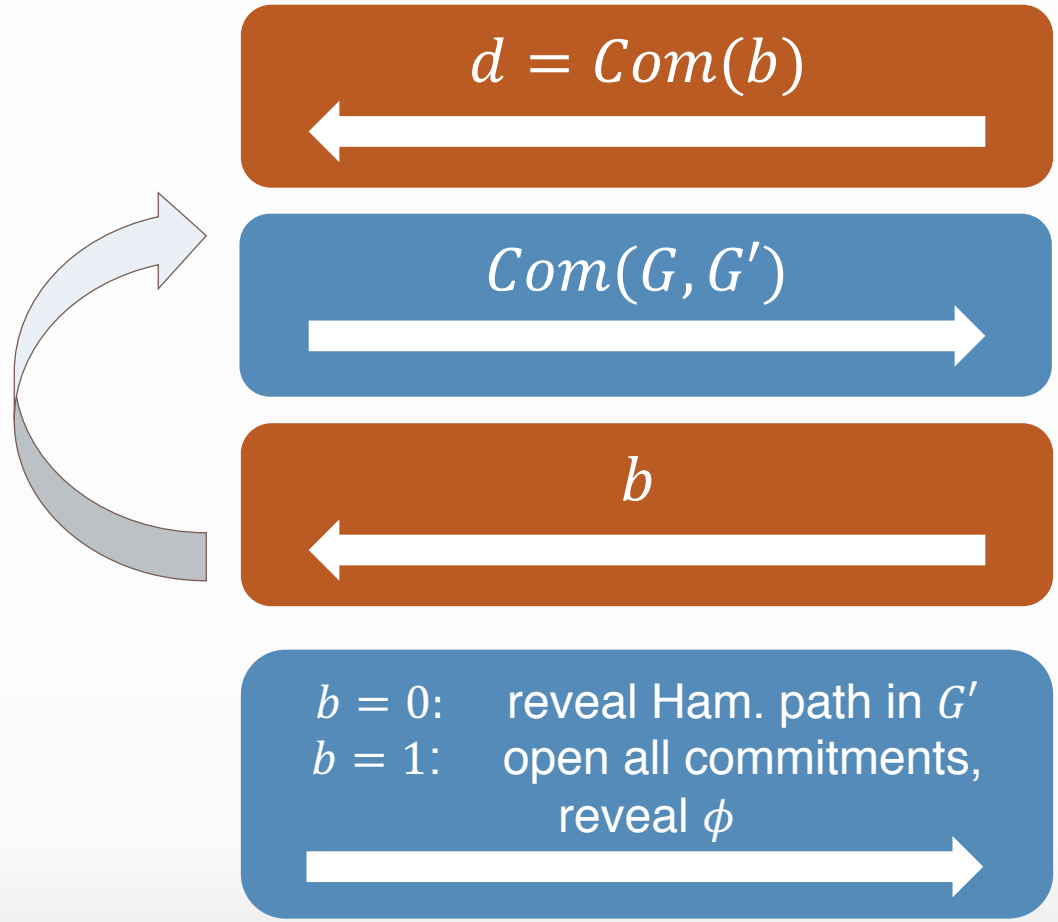
$\mathbb{E}[time(S)] = 2^k time(V^*)$

Idea: Use commitments to get around the impossibility result

Parallel HAM



HVZK parallel HAM



HVZK intuition

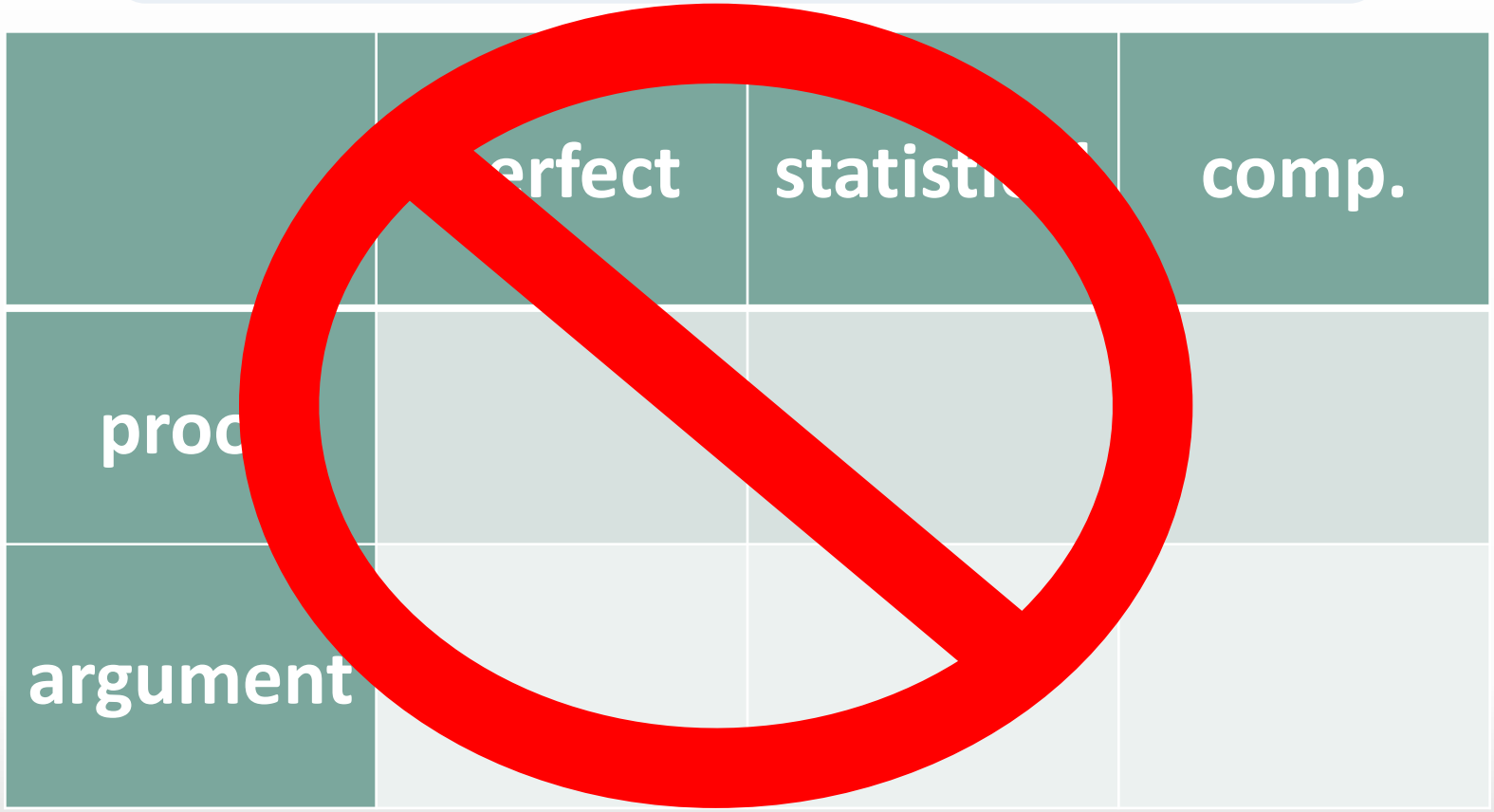
- Rewind V to time d is known
- V cannot change their commitment

Only **argument**

Com has to be stat. binding, thus can be only comp. hiding

Different flavours of ZK

Interactive vs Non-interactive



ZK applications

Schnorr identification scheme

Knows w such that $x = g^w$

$k \in_R \mathbb{Z}_p$, send I $= g^k$

$r \in_r \mathbb{Z}_p$

$s = rw + k \bmod p$

ACCEPT iff $g^s x^{-r} = I$

x

Note: For cyclic $\mathbb{G}$ for all x there is w st $x = g^w$
Thus the language **is trivial**
However Andrew shows here that **he knows** w

Not Zero-Knowledge

honest verifier zero-knowledge

Special soundness:
Given two conversations (I, r, s) and (I, r', s') such that $r \neq r'$one can extract witness w

Proof:

$$I = g^k, s = rw + k, s' = r'w + k$$
$$g^s x^{-r} = I = g^{s'} x^{-r'}$$
$$g^{s-s'} = x^{-r'+r}$$
$$g^{(s-s')/(-r'+r)} = x$$
$$s, s', r, r' \textbf{ known!}$$
$$\boldsymbol{w = (s - s')/(r - r')}$$

Remember: the language is trivial

Proofs vs Proofs of Knowledge

Proof:
Prover shows $x \in \mathcal{L}$

Proof of Knowledge:
Prover shows $x \in \mathcal{L}$ and he knows a witness w for that

What does it mean to know?

- Informally: We can make P output w
- More formally: There exists machine E called extractor that can output w after interacting with P

Extractor works like a simulator but doesn't interact with V but with P

How many rounds?

4-round protocols for $\mathbb{NP}$ exist

If for language L exists a ZK protocol with 3 rounds and negligible soundess then L is trivial ($\mathcal{L} \in \mathbb{BPP}$)

Non-interactive, 1-round protocol **impossible**

We will **show** them during the next lecture

Thank you!

Part I ends here, stay tuned for the Part II